



UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA



UNIVERSITAT POLITÈCNICA DE VALÈNCIA

Escuela Técnica Superior de Ingeniería Informática

Desarrollo de una aplicación para visualizar y analizar
imágenes hiperespectrales

Trabajo Fin de Grado

Grado en Ingeniería Informática

AUTOR/A: García Achiakhi, Luis

Tutor/a: Albert Gil, Francisco Eugenio

Cotutor/a: Rodríguez Ortega, Alejandro

CURSO ACADÉMICO: 2023/2024

Agradecimientos

A Alejandro

A Francisco

A Marc

A Campanilla

A Rogeta

Resumen

Actualmente los sistemas de adquisición de imágenes hiperespectrales están muy demandados en diferentes campos de investigación, como son la medicina, la farmacología, alimentación, control de calidad, teledetección, etc. Estos sistemas de imagen permiten adquirir información de diferentes modalidades espectrales (NIR, VIS, UV, IR, etc.) para cada uno de los píxeles en los que se divide una muestra, obteniendo así información espectral y espacial, lo que lo convierte en una herramienta muy interesante para multitud de aplicaciones.

En el mercado existen muchas aplicaciones comerciales que permiten analizar este tipo de imágenes y obtener ciertos resultados. Sin embargo, estos programas tienen un elevado coste y su uso suele requerir de una formación previa del usuario final. Por otro lado, las aplicaciones de acceso en abierto para el análisis de este tipo de imágenes que se adapten a cualquier campo de investigación son difíciles de encontrar y muchas veces se recurre a la compra de estas adaptadas a una necesidad concreta, con el siguiente sobrecoste

El Trabajo de Fin de Grado consistirá en el desarrollo de una interfaz gráfica de usuario (GUI) sencilla y amigable, basada en lenguaje Python, que permita la visualización y el análisis de imágenes hiperespectrales de diferentes disciplinas, que además de minimizar la curva de aprendizaje del usuario, permita la integración de nuevos desarrollos realizados por otros investigadores.

En esta primera aproximación se pretenden las siguientes características:

- Fácil navegación e interacción con el hipercubo, a través de cursores enlazados con la navegación y la representación de diferentes gráficos que ayuden a la interpretación visual de las imágenes bajo estudio.
- Soporte de diferentes formatos de imágenes hiperespectrales que se puedan encontrar en el mercado.
- Implementación del proceso de segmentación, tanto manual como semiautomático, de las regiones de interés en todo el espectro disponible en el fichero.
- Implementación de preprocesado espectral y espacial, como filtros, operaciones morfológicas, eliminación de regiones, etc.
- Exportación de los resultados en diversos formatos utilizados en el ámbito científico y almacenamiento de las imágenes procesadas.

Resum

Actualment els sistemes d'adquisició d'imatges hiperespectrals estan molt demandats en diferents camps d'investigació, com són la medicina, la farmacologia, alimentació, control de qualitat, teledetecció, etc. Aquests sistemes d'imatge permeten adquirir informació de diferents modalitats espectrals (NIR, VIS, UV, IR, etc.) per a cada un dels píxels en els quals es dividix una mostra, obtenint així informació espectral i espacial, la qual cosa la converteix en una ferramenta molt interessant per a multitud d'aplicacions.

En el mercat existixen moltes aplicacions comercials que permeten analitzar aquest tipus d'imatge i obtenir uns certs resultats. No obstant això, aquests programes tenen un elevat cost i el seu ús sol requerir d'una formació prèvia de l'usuari final. D'altra banda, les aplicacions d'accés en obert per a l'anàlisi d'aquest tipus d'imatges que s'adapten a qualsevol camp d'investigació són difícils de trobar i moltes vegades es recorre a la compra d'una adaptada a una necessitat concreta, amb el consegüent sobrecost.

El Treball de Fi de Grau consistirà en el desenvolupament d'una interfície gràfica d'usuari (GUI) senzilla i amigable, basada en llenguatge Python, que permeti la visualització i anàlisi d'imatges hiperespectrals de diferents disciplines, que a més de minimitzar la corba d'aprenentatge de l'usuari, permeti la integració de nous desenvolupaments realitzats per altres investigadors.

En aquesta primera aproximació es pretén obtenir les següents característiques:

- Fàcil navegació i interacció amb el hipercub, a través de cursors enllaçats amb la navegació i la representació de diferents gràfics que ajuden a la interpretació visual de les imatges sota estudi.
- Suport de diferents formats d'imatges hiperespectrals que es puguin trobar en el mercat.
- Implementació del procés de segmentació, tant manual com semiautomàtic, de les regions d'interés en tot l'espectre disponible en el fitxer.
- Implementació de preprocessament espectral i espacial, com a filtres, operacions morfològiques, eliminació de regions, etc.
- Exportació dels resultats en diversos formats utilitzats en l'àmbit científic i emmagatzematge de les imatges processades.

Abstract

Nowadays, hyperspectral imaging systems are highly demanded in different research fields, such as medicine, pharmacology, food, quality control, remote sensing, etc. These imaging systems allow acquiring information from different spectral modalities (NIR, VIS, UV, IR, etc.) for each of the pixels into which a sample is divided, thus obtaining spectral and spatial information, which makes it a very interesting tool for a multitude of applications.

There are many commercial applications on the market that can be used to analyze this type of images and obtain certain results. However, these programs have a high cost and their use usually requires prior training by the end user. On the other hand, it is difficult to find open access applications for the analysis of this type of images that can be adapted to any field of research, and often these applications are purchased adapted to a specific need, with the resulting additional cost.

The Final Degree Project will consist of the development of a simple and user-friendly graphical user interface (GUI), based on Python language, that allows the visualization and analysis of hyperspectral images from different disciplines, which in addition to minimizing the user's learning curve, allows the integration of new developments made by other researchers.

In this first approach the following features are intended:

- Easy navigation and interaction with the hypercube, through cursors linked to navigation and the representation of different graphics that help the visual interpretation of the images under study.
- Support of different hyperspectral image formats that can be found in the market.
- Implementation of the segmentation process, both manual and semi-automatic, of the regions of interest in the entire spectrum available in the file.
- Implementation of spectral and spatial preprocessing, such as filters, morphological operations, region elimination, etc.
- Export of the results in various formats used in the scientific field and storage of the processed images.

Palabras clave

Imagen Hiperespectral; Agroingeniería; lenguaje Python; Espectro electromagnético; Interfaz gráfica; Procesado de imágenes.

1. Índice

1. Índice.....	6
2. Índice de Figuras.....	8
3. Introducción.....	9
3.1 Motivación.....	10
3.2 Estructura de la memoria	11
3.3 Objetivos	12
4. Colaboraciones	13
5. Estado del arte	14
5.1 Conceptos básicos de la imagen hiperespectral	14
5.1.1 Radiancia.....	14
5.1.2 Reflectancia.....	15
5.1.3 Espectro electromagnético.....	16
5.1.4 Imagen hiperespectral.....	17
5.1.5 Imagen de falso color	19
5.2 Aplicaciones comerciales	20
5.2.1 Hyperspectral viewer:.....	21
5.2.2 SpecimINSIGHT:	23
5.2.3 HYPER-Tools:	24
5.3 Crítica al estado del arte.....	25
6. Análisis de requisitos	26
6.1 Interesados en el desarrollo (stakeholders).....	26
6.2 Personas	27
6.3 Escenarios de uso	29
6.4 Requisitos.....	30
7. Situación de desarrollo y mercado	31
7.1 Ventajas del proyecto.....	31
7.2 Limitaciones	32
8. Diseño de la solución.....	33
8.1 Arquitectura del sistema	33
8.2 Materiales.....	33
9. Proceso de desarrollo	36

9.1 Sprint 1	37
9.2 Sprint 2	39
9.3 Sprint 3	40
9.4 Sprint 4	42
9.5 Sprint 5	44
9.6 Sprint 6	46
9.7 Sprint 7	49
9.8 Sprint 8	52
10. Conclusiones	55
11. Trabajo futuro	56
12. Relación del proyecto desarrollado con los estudios realizados	57
13. Bibliografía	58
14. ANEXO 1	61
OBJETIVOS DE DESARROLLO SOSTENIBLE.....	61

2. Índice de Figuras

Figura 1 Varias bandas de una misma imagen hiperespectral, y la imagen en falso color de esa imagen	9
Figura 2 Fórmula de la radiancia [1]	14
Figura 3 Imagen de referencia sobre el concepto de la radiancia [1]	15
Figura 4 Muestra de las diferencias en reflectancia entre diferentes materiales [2]	15
Figura 5 Representación del espectro electromagnético [3].....	16
Figura 6 Representación del espectro visible por el ojo humano [4].....	17
Figura 7 Imagen de referencia para definir el concepto de imágenes hiperespectrales [5]	18
Figura 8 Representación de un hipercubo [6].....	18
Figura 9 Imagen en color falso	19
Figura 10 Captura de la interfaz de Hyperspectral viewer obtenida de su página web [7] .	21
Figura 11 Captura de la interfaz de Hyperspectral Viewer.....	22
Figura 12 Captura de la interfaz de usuario de SpecimINSIGHT [8]	23
Figura 13 Interfaz de la aplicación HYPER-Tools [9].....	24
Figura 14 Interfaz de la aplicación HYPER-Tools [9].....	24
Figura 15 IVIA [10]	26
Figura 16 Captura de la selección de imagen de la aplicación.....	37
Figura 17 Captura de la primera versión de la aplicación	38
Figura 18 Captura de la segunda versión de la aplicación	39
Figura 19 Captura de la interfaz de la tercera versión de la aplicación, en la banda 0	40
Figura 20 Captura de la interfaz de la tercera versión de la aplicación, en la banda 30	40
Figura 21 Captura de la aplicación mostrando el gráfico generado al seleccionar un punto de la imagen	41
Figura 22 Captura de la aplicación que muestra la funcionalidad de falso color	42
Figura 23 Captura de la selección de imagen de la aplicación.....	43
Figura 24 Captura de la funcionalidad de selección de un área rectangular de la aplicación	44
Figura 25 Captura de la funcionalidad de selección de un área rectangular de la aplicación	45
Figura 26 Captura de la funcionalidad de selección de un área rectangular de la aplicación	45
Figura 27 Captura del resultado de la selección de longitudes de onda de la aplicación .	46
Figura 28 Captura del resultado de la selección de longitudes de onda de la aplicación .	47
Figura 29 Captura del resultado de la selección de longitudes de onda de la aplicación .	47
Figura 30 Captura de la búsqueda de imágenes de la aplicación.....	48
Figura 31 Captura de la búsqueda de imágenes de la aplicación.....	48
Figura 32 Captura de la primera función a incluir.....	49
Figura 33 Captura de la segunda función a incluir.....	50
Figura 34 Captura del código de la aplicación	50
Figura 35 Imágenes resultado de las funciones introducidas	51
Figura 36 Muestra visual del proceso de normalización	52
Figura 37 Banda 0 de la imagen de referencia blanca.....	53
Figura 38 Captura de la selección del área de la imagen de referencia blanca	54
Figura 39 Captura de la aplicación tras la realización de la normalización	54

3. Introducción

Los sistemas de manipulación de imágenes hiperespectrales están en un punto muy alto de demanda en varios campos de investigación. Estos sistemas ofrecen la posibilidad de obtener información de varias modalidades espectrales para cada píxel de una muestra, haciendo de ellos una herramienta importante para muchos campos de investigación.

Existen muchas aplicaciones comerciales dedicadas a permitir trabajar con este tipo de imágenes, pero todas tienen multitud de limitaciones, incluyendo un precio elevado y la necesidad de ser usadas por un usuario especialmente formado. Además, la posibilidad de añadir o modificar funcionalidades a la aplicación está limitada a sus desarrolladores, forzando a los usuarios a conformarse con las funcionalidades por defecto, o a comprar aplicaciones adaptadas a una necesidad concreta, generando un sobrecoste.

El objetivo de este Trabajo de Fin de Grado es el desarrollo de una interfaz gráfica de usuario (GUI) cómoda, basada en Python, que dé la posibilidad de visualizar y trabajar con imágenes hiperespectrales, reduzca la curva de aprendizaje del usuario y permita integrar nuevos desarrollos por parte de otros investigadores.

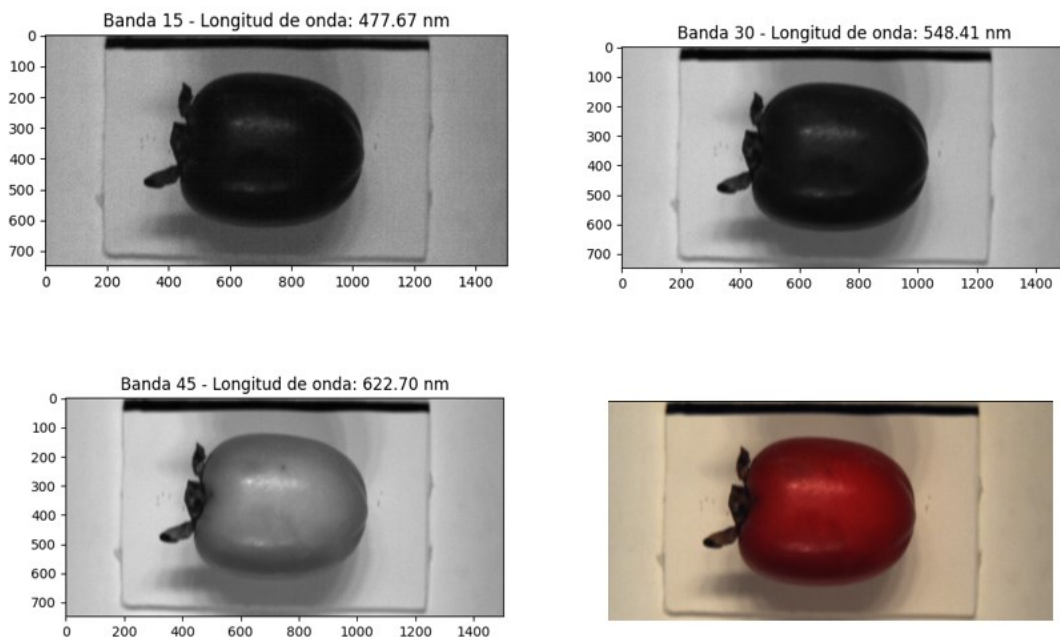


Figura 1 Varias bandas de una misma imagen hiperespectral, y la imagen en falso color de esa imagen

3.1 Motivación

Existen en el mercado varias aplicaciones para procesar y analizar imágenes hiperespectrales, pero la mayoría de ellas exigen de un pago para su uso, y son de código cerrado, por lo que sólo permiten hacer aquello que su desarrollador o comercializador deseen, quedando estas aplicaciones limitadas.

Por estas razones, surge la necesidad de una aplicación que ofrezca las funcionalidades básicas de trabajo con imágenes, con el añadido de ser de libre acceso, por lo que no sería necesario efectuar ningún pago para su uso, y de código abierto, para permitir añadir o modificar funcionalidades que el usuario pueda desear a posteriori.

El Instituto Valenciano de Investigaciones Agrarias (IVIA) tenía especial interés en el desarrollo de una herramienta semejante, pues sus investigadores se ven obligados a trabajar con herramientas que no satisfacen por completo sus necesidades, y que les obligan a plegarse a multitud de limitaciones, por lo que, al establecerse el contacto entre esta organización, y el estudiante a cargo de este proyecto, surgió la oportunidad de trabajar en una herramienta más adaptada a sus necesidades.

3.2 Estructura de la memoria

En primer lugar, se definen los objetivos que se busca cumplir con la aplicación a desarrollar.

Posteriormente, se realiza un análisis del estado del arte, repasando las nociones físicas de las imágenes hiperespectrales, las aplicaciones que hay en el mercado, y cómo competir con ellas.

Después se realiza un análisis de requisitos, en que se emplean varias técnicas para descubrir cuáles son las necesidades de la aplicación a desarrollar.

En el apartado siguiente se realiza un estudio de la situación del desarrollo del proyecto y su competitividad.

Tras ese apartado, se define el diseño de la solución, y se entra en detalles básicos sobre la arquitectura y tecnología usadas.

A continuación, se explica el proceso de desarrollo de la aplicación, separado por los periodos de desarrollo, o sprints, en que se dividió.

Tras esto, se exponen las conclusiones alcanzadas, así como una reflexión sobre el trabajo futuro, y la relación del proyecto con el grado.

Finalmente, el último apartado de la memoria contiene la bibliografía, con la información obtenida de fuentes ajenas correctamente documentada.

3.3 Objetivos

Los objetivos con el desarrollo de esta aplicación son los siguientes:

- Debe ofrecer una versión funcional de la mayoría de los requisitos que se definan para la aplicación final. Se pretende realizar una primera versión que ofrezca una experiencia suficientemente completa para el usuario.
- Debe tener una curva de aprendizaje lo más favorable posible tanto para personal especializado como para nuevos usuarios no familiarizados con la materia.
- Debe ser de código abierto y permitir introducir funciones nuevas y modificar las ya existentes con facilidad.

4. Colaboraciones

El desarrollo de la aplicación ha sido individual, pero se ha dispuesto de ayuda por parte del Instituto Valenciano de Investigaciones Agrarias (IVIA).

Este centro lleva años trabajando con imágenes hiperespectrales y desarrollando sistemas de adquisición de estas, por lo que sus investigadores pueden considerarse expertos en la materia.

El IVIA ofreció al alumno del TFG un recorrido por sus instalaciones, mostrando directamente cómo funcionan los sistemas de adquisición de imágenes, y qué uso hacen de ellas, y durante el desarrollo, facilitaron el acceso a multitud de imágenes hiperespectrales con las que poder realizar pruebas.

5. Estado del arte

Esta sección servirá para revisar las bases físicas sobre las que se sustentan las imágenes hiperespectrales, así como para explorar otras aplicaciones informáticas, que pasarán a ser la competencia directa de esta aplicación una vez esté finalizada, para permitir después realizar un análisis de las mismas y ver cómo pueden mejorarse.

5.1 Conceptos básicos de la imagen hiperespectral

Antes de comenzar a describir el propio proyecto, es importante tener unos fundamentos físicos básicos sobre las imágenes hiperespectrales. A continuación, se ofrece información sobre algunos de los conceptos de mayor importancia:

5.1.1 Radiancia

Se refiere a la cantidad de flujo radiante, o energía por unidad de tiempo, medida en vatios (W), que incide en un sistema, teniendo en cuenta la longitud de onda, medida en micras (μm), el cono por unidad de ángulo sólido en que incide, medido en estereorradianes (sr), y el área, medida en metros cuadrados (m^2). La fórmula es la siguiente:

$$L_e = \frac{\partial^2 \Phi(\lambda)}{\partial A_s \partial \Omega} \frac{W}{\text{m}^2 \mu\text{m sr}} \quad (1)$$

Figura 2 Fórmula de la radiancia [1]

La Figura 3 permite visualizar mejor el concepto:

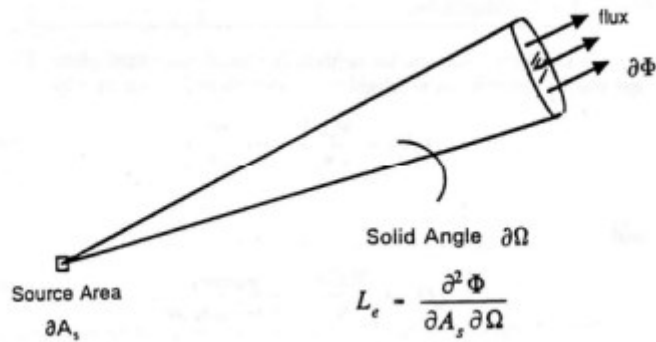


Figura 3 Imagen de referencia sobre el concepto de la radiancia [1]

5.1.2 Reflectancia

Se define esta magnitud como la relación entre la intensidad de la radiación que refleja una muestra y la intensidad de la radiación incidente en esta muestra, es decir, que la reflectancia se puede calcular con la siguiente fórmula:

$$\text{Reflectancia} = \frac{\text{Radiación Reflejada}}{\text{Radiación Incidente}} \quad (2)$$

Cada material produce un espectro de reflectancia distinto, como se puede ver a continuación:

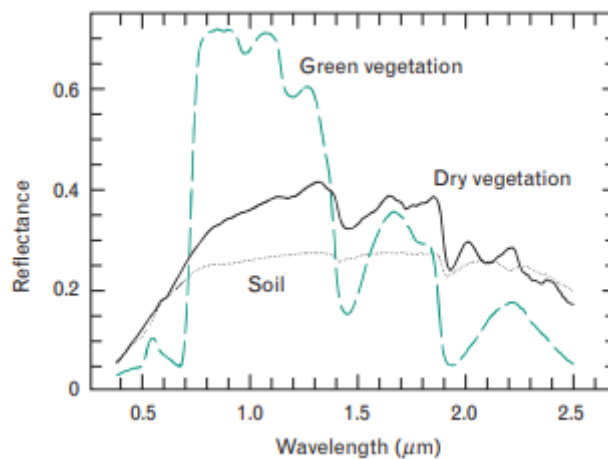


Figura 4 Muestra de las diferencias en reflectancia entre diferentes materiales [2]

Con la reflectancia, se obtiene la firma espectral, la cual es única para cada material, por lo que, a través de la reflectancia, es posible conocer la composición de cualquier objeto que se capte, o identificar objetos que se sabe que están compuestos por determinado material.

5.1.3 Espectro electromagnético

El espectro electromagnético es el rango completo de todas las frecuencias posibles en las que la radiación se puede transmitir en ondas. La radiación electromagnética se compone de ondas de energía que viajan a través del espacio y pueden variar en longitud de onda y frecuencia (Figura 5). El espectro electromagnético incluye desde las ondas de radio, con longitudes de onda largas y frecuencias bajas, hasta los rayos gamma que tienen longitudes de onda muy corta y frecuencia muy altas.

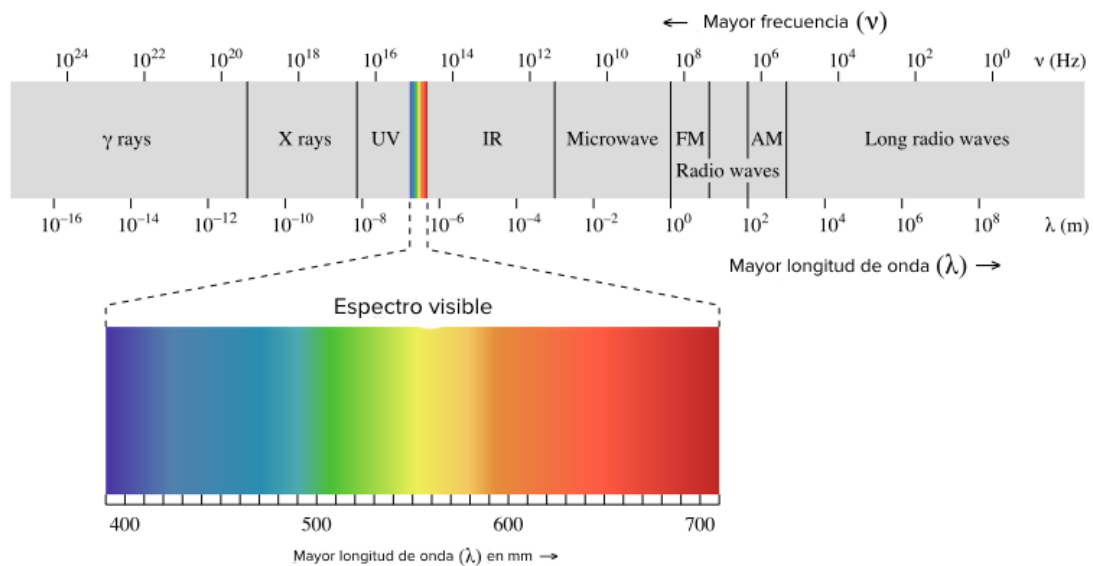


Figura 5 Representación del espectro electromagnético [3]

Como puede verse, la radiación perceptible por los ojos humanos (Figura 6), usualmente llamada espectro visible, es extremadamente limitada, abarcando las longitudes de onda que van desde los 400 nanómetros (nm) hasta los 700 nm. Dentro de este rango las diferentes longitudes de onda se perciben como colores distintos:

- Azul: $\sim 450 - 495$ nm.
- Verde: $\sim 495 - 570$ nm.
- Rojo: $\sim 620 - 700$ nm.

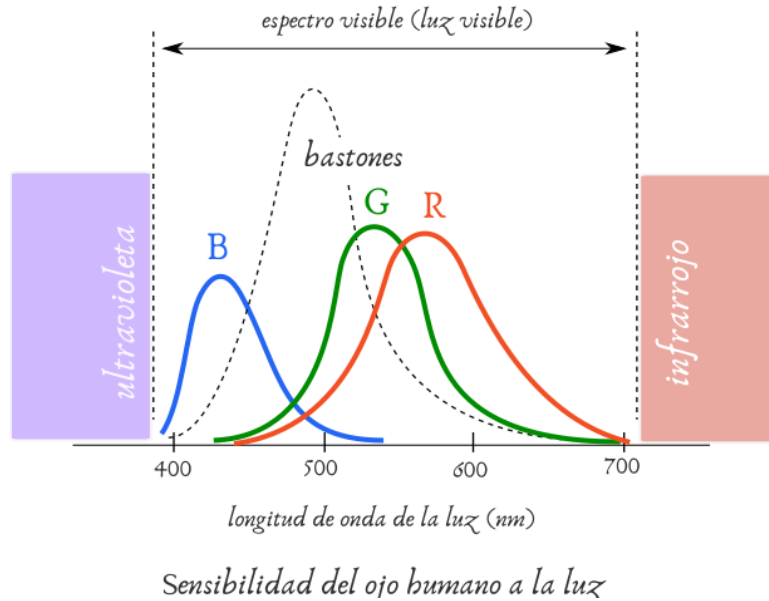


Figura 6 Representación del espectro visible por el ojo humano [4]

Las imágenes hiperespectrales trabajan no solo en el espectro visible, sino también en otras regiones del espectro electromagnético que no son visibles para el ojo humano. Estas regiones incluyen los rangos de:

- Ultravioleta cercano: ~10 – 400 nm
- Visible: ~400 – 700 nm
- Infrarrojo cercano: 700 – 2500 nm
- Infrarrojo 2500nm – 1 mm

Esto hace que las imágenes hiperespectrales, que permiten visualizar algunas longitudes de onda que el ojo humano no percibe, sean una gran ayuda en varios sectores, como el médico o el agrario, en que esas longitudes pueden servir para identificar todo tipo de problemas.

5.1.4 Imagen hiperespectral

La imagen hiperespectral es una técnica de captura de imágenes que recoge y procesa información de todo el espectro electromagnético. A diferencia de las imágenes tradicionales que capturan la información en tres bandas o canales (rojo, verde y azul), las imágenes hiperespectrales capturan un número mayor de bandas estrechas en diferentes longitudes de onda, que pueden ir desde el espectro ultravioleta hasta el infrarrojo, lo cual permite identificar, cuantificar y analizar con

gran precisión la composición y las propiedades de los materiales bajo estudio.

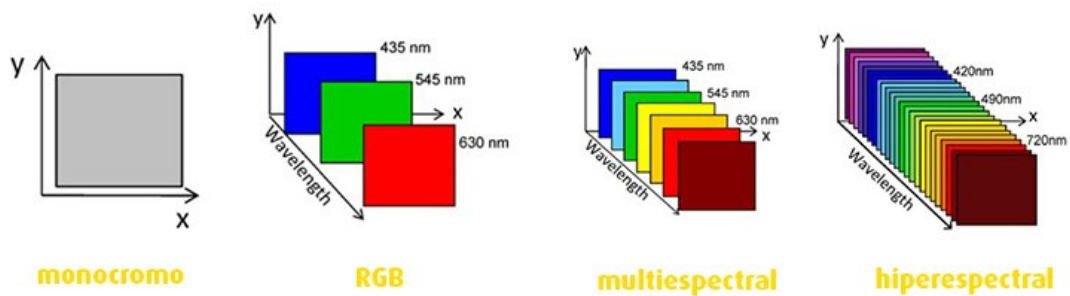


Figura 7 Imagen de referencia para definir el concepto de imágenes hiperespectrales [5]

El resultado de esta captura es un conjunto de datos tridimensionales conocido como hipercubo. Este hipercubo se estructura en dos dimensiones espaciales (alto y ancho de la imagen) y una dimensión espectral, las diferentes longitudes de onda capturadas por cada píxel. De este modo, cada píxel en la imagen hiperespectral no es simplemente un punto de color, sino un vector espectral completo que forma parte del hipercubo, integrando toda la información espacial y espectral.

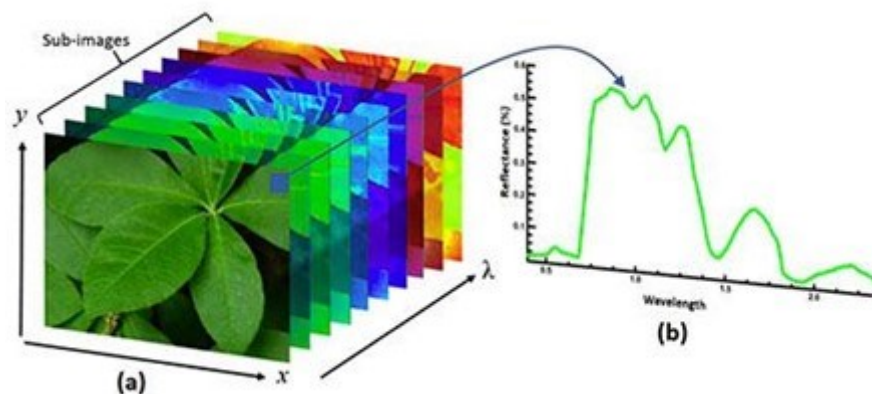


Figura 8 Representación de un hipercubo [6]

Las imágenes hiperespectrales y el análisis y procesado del hipercubo tienen una amplia gama de aplicaciones en diversas áreas debido a su capacidad para identificar y caracterizar materiales con gran precisión. Algunos de los campos de investigación en que se usan imágenes hiperespectrales incluyen desde la agricultura de precisión para detectar plagas y enfermedades de las plantas y cultivos, a la medicina, donde la imagen hiperespectral es utilizada para realizar diagnósticos no invasivos de tejidos enfermos.

El procesamiento de imágenes hiperespectrales requiere software especializado capaz de manejar y analizar los grandes volúmenes de datos generados por estas técnicas. Estos programas están diseñados para extraer información relevante del hipercubo y permitir a los usuarios poder evaluar e interpretar las imágenes adquiridas.

En el siguiente apartado, se detallarán ejemplos de aplicaciones software comerciales que trabajan con imágenes hiperespectrales.

5.1.5 Imagen de falso color

Una imagen en falso color es una representación de una imagen hiperespectral, en la cual se emplea la información de tres longitudes de onda extraídas del hipercubo, asignando cada una a los canales de color rojo, verde y azul (RGB) para generar una imagen a color. Generalmente, se seleccionan longitudes de onda que coinciden con los colores primarios (Rojo - 685 nm, Verde - 533 nm y Azul - 472 nm), aunque en realidad podría utilizarse cualquier otra. Es importante destacar que esta representación no constituye una reproducción precisa de los colores reales del objeto, ya que para cada color se selecciona una única banda espectral en lugar de la información correspondiente al rango espectral completo del color. Por esta razón, la apariencia resultante no coincide totalmente con la percibida en la realidad, lo que justifica el término "falso color" para describir este tipo de reconstrucción.



Figura 9 Imagen en color falso

5.2 Aplicaciones comerciales

En la siguiente sección se busca conocer otras aplicaciones similares del mercado para comprobar y aprender sobre su funcionamiento, interfaz y facilidades para el usuario, lo que permitirá mejorar algunos de estos aspectos con el objetivo de ofrecer la mejor aplicación posible.

En este proyecto, esta investigación ha sido especialmente compleja, pues como ya se ha indicado, las aplicaciones suelen estar muy especializadas, o ser de pago. Sin embargo, se ha podido ver el funcionamiento de algunas de estas aplicaciones en ordenadores ajenos, y explorar las páginas de algunas de estas aplicaciones ofrece suficiente información sobre ellas.

A continuación, se revisarán las funcionalidades principales de la aplicación que se ha considerado más interesante para este proyecto, cuya interfaz es la principal referencia para la del proyecto:

5.2.1 Hyperspectral viewer:

Esta aplicación, desarrollada por la compañía Mathworks para MATLAB, una plataforma de programación y cálculo numérico, es especialmente importante, y se puede aprender mucho sobre el diseño y las funciones de este tipo de aplicaciones simplemente observando su uso.

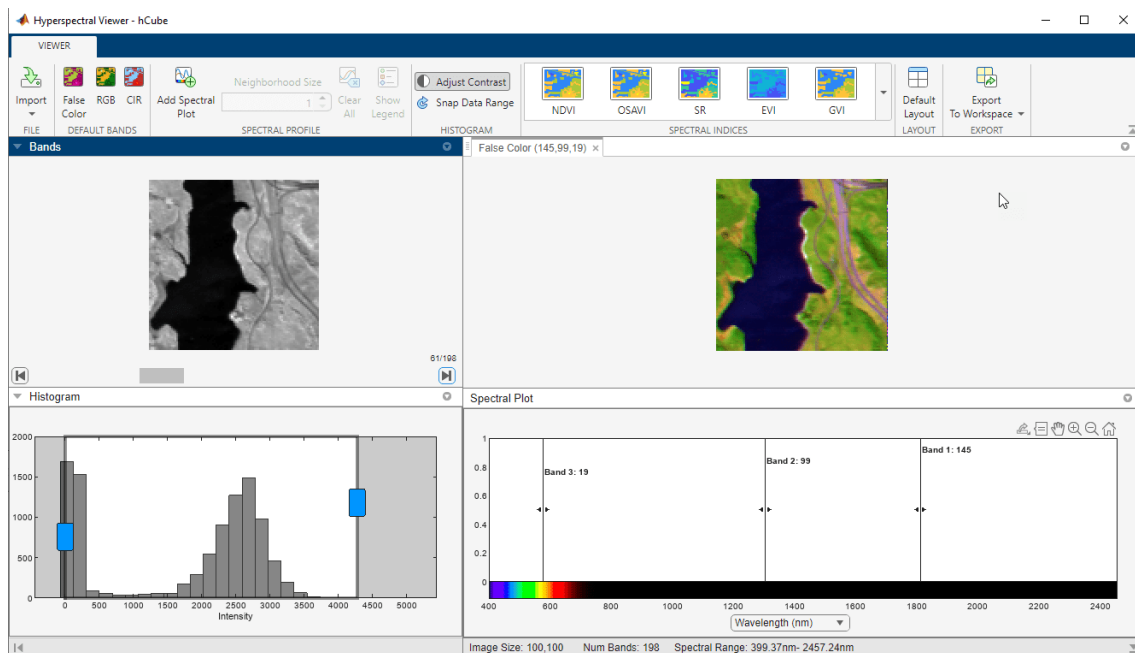


Figura 10 Captura de la interfaz de Hyperspectral viewer obtenida de su página web [7]

Hyperspectral Viewer permite visualizar este tipo de imágenes, pero también ofrece otras dos funcionalidades de gran importancia que, como se verá en la especificación de requisitos, se consideran prioritarias en el desarrollo de este tipo de aplicación:

- La imagen en falso color, ubicada en la esquina superior derecha de la captura, que ofrece una copia de la imagen hiperespectral tratada, pero con un aspecto cercano al que tendría en la realidad, al usar la información ofrecida por la propia imagen.
- En la esquina inferior izquierda se muestra el histograma de la imagen en la banda seleccionada en escala de grises, el cual cambia al seleccionar cualquier imagen dentro del hipercubo.

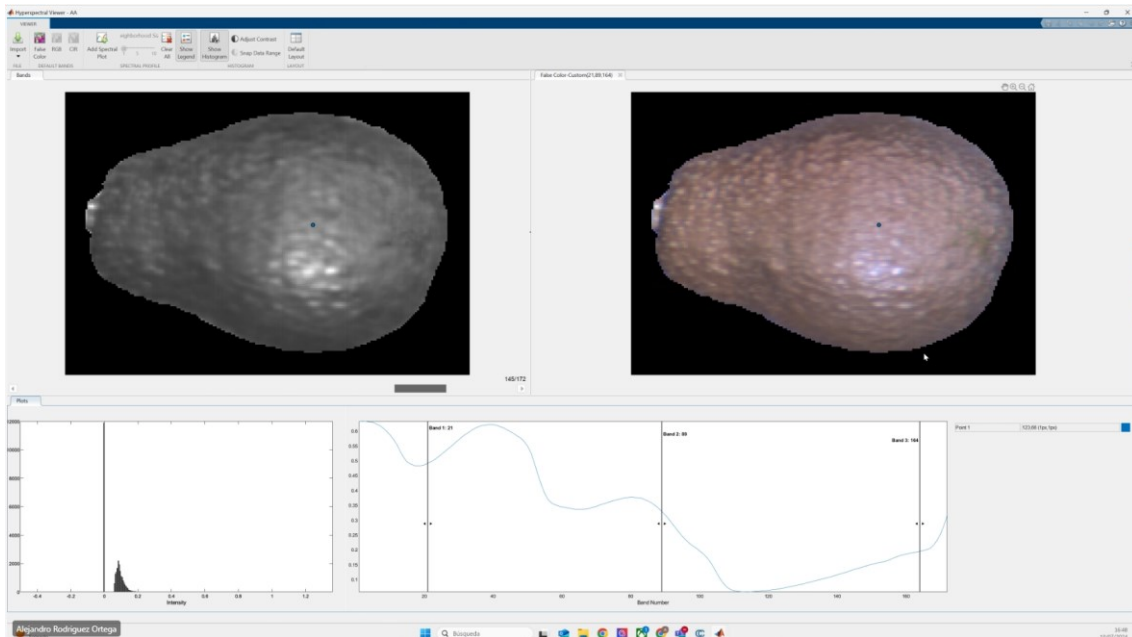


Figura 11 Captura de la interfaz de Hyperspectral Viewer

La ventana inferior derecha permite visualizar la información relativa a las variaciones en el espectro electromagnético.

También son importantes las tres líneas que aparecen a lo largo del gráfico. Estas líneas representan las longitudes de onda que se emplean como referencia para los tres colores primarios en la simulación de color, y como se puede ver, puede modificarse su posición, lo que hace que se refleje una combinación de colores distinta como resultado.

5.2.2 SpecimINSIGHT:

SpecimINSIGHT es una aplicación de procesamiento de imágenes hiperespectrales que ofrece multitud de herramientas para trabajar con estas, incluyendo funcionalidades ya vistas en Hyperspectral Viewer, como la que ofrece una imagen de falso color, o la que genera un gráfico con las variaciones en el espectro electromagnético, pero esta herramienta ofrece también funcionalidades no vistas en la anterior, y permite incluso seleccionar varias áreas de la imagen, dibujadas a mano, para obtener datos sobre el espectro electromagnético en esas secciones de la imagen.

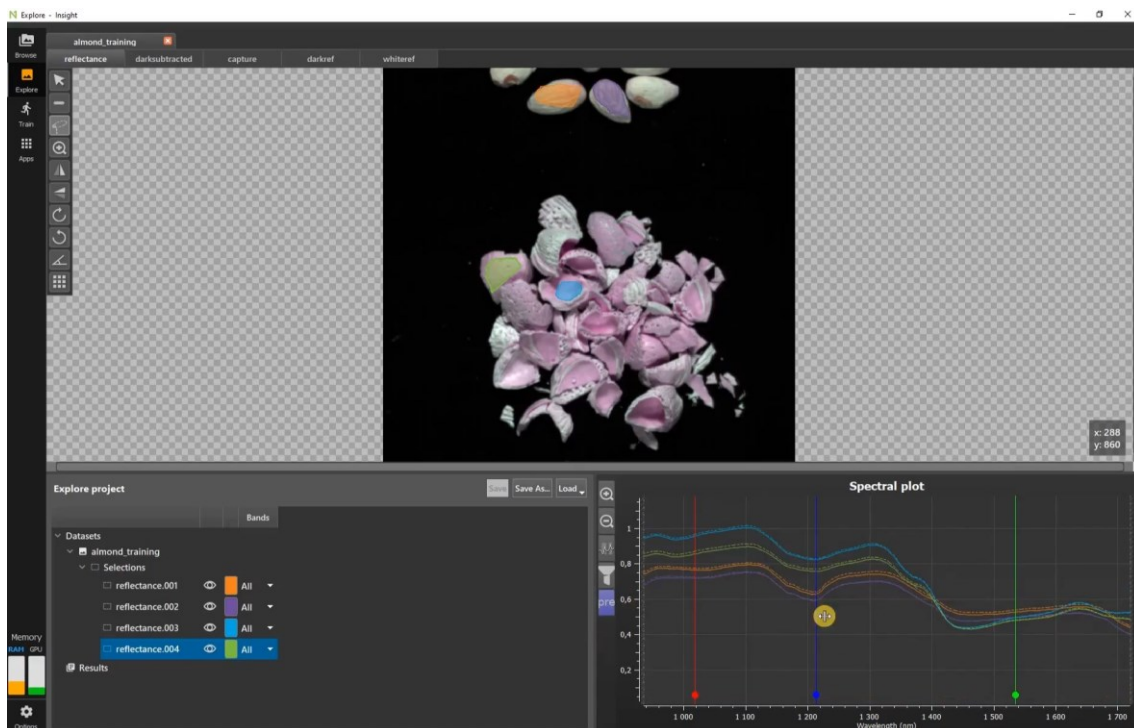


Figura 12 Captura de la interfaz de usuario de SpecimINSIGHT [8]

Esta es, sin embargo, una aplicación muy especializada, diseñada para funcionar con los sets de datos que ofrecen los dispositivos de la empresa specim, lo que hace que su uso para un usuario medio sea muy complicado.

5.2.3 HYPER-Tools:

Esta es una aplicación hecha en Matlab que ofrece las funciones básicas de este tipo de aplicaciones, como la imagen en falso color, o la generación de gráficos.

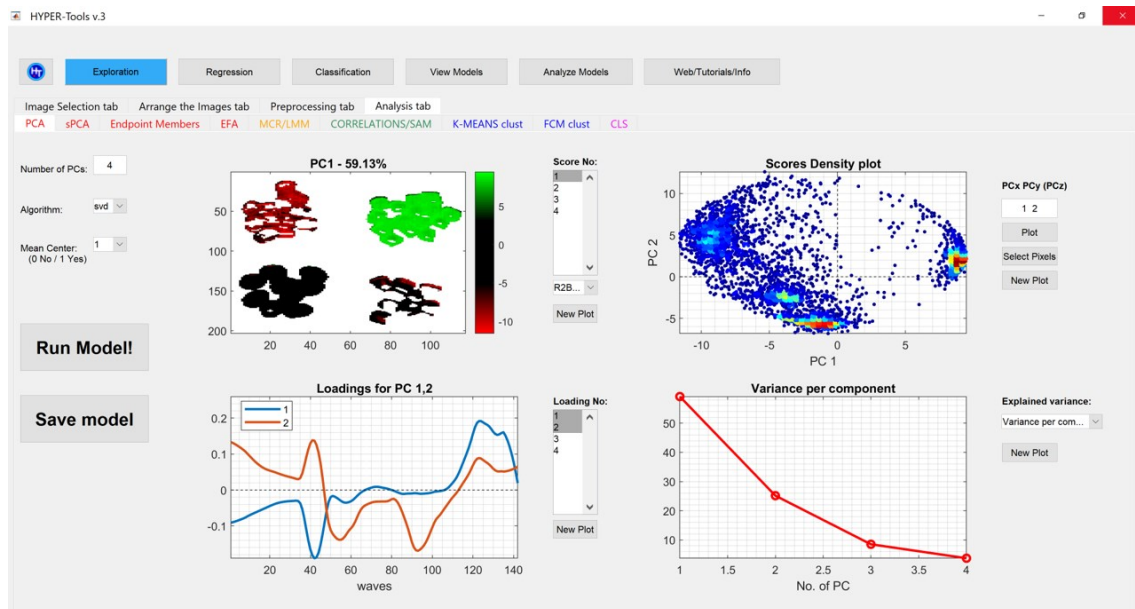


Figura 13 Interfaz de la aplicación HYPER-Tools [9]

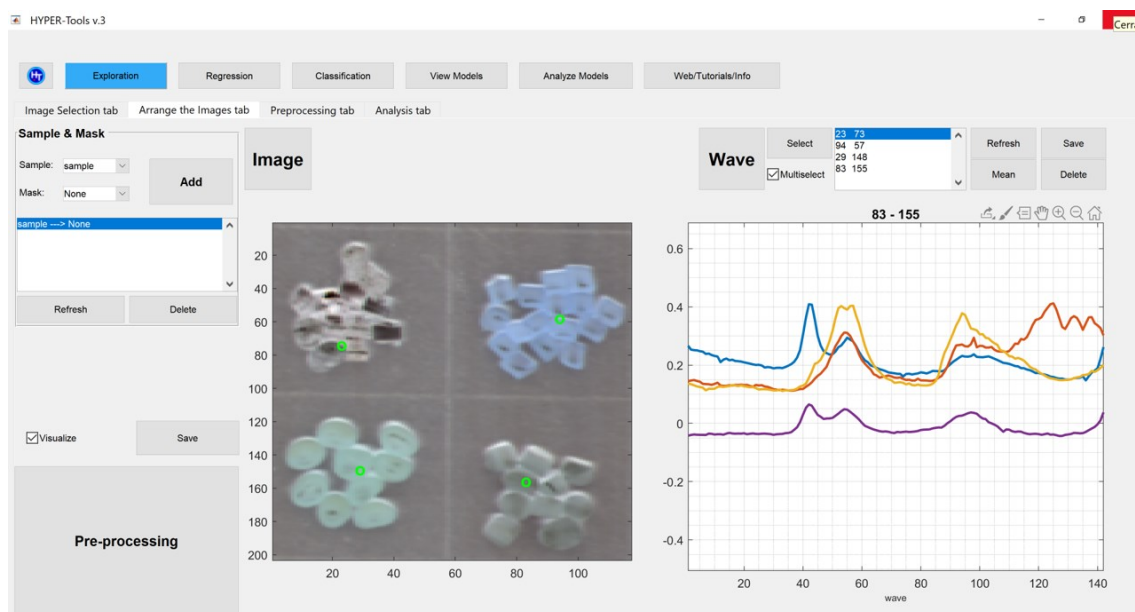


Figura 14 Interfaz de la aplicación HYPER-Tools [9]

Esta aplicación no ofrece ninguna nueva funcionalidad en que inspirarse, pero su interfaz resulta interesante como posible punto de partida para este proyecto, antes de crear un menú más elaborado para los usuarios.

5.3 Crítica al estado del arte

Al revisar cualquier aplicación relacionada con el tratamiento de imágenes hiperespectrales, comienzan a surgir multitud de problemas, no necesariamente con su funcionamiento, sino con la experiencia de usuario que ofrecen:

- Estas aplicaciones, en muchos casos bien diseñadas, están totalmente limitadas. No existe posibilidad alguna de introducir funcionalidades nuevas o de modificar las ya presentes.
- Estas aplicaciones son, en algunos casos, excesivamente caras, por lo que su uso no es recomendable para nadie que no sea un experto en la materia que sepa que va a darle un uso prolongado en el tiempo
- Estas aplicaciones solo se actualizan tanto como su desarrolladora o distribuidora lo deseen. Si no se introduce una función que el usuario desea, no tiene forma de añadirla él mismo.

Todos estos son problemas que la mayoría de aplicaciones de este tipo están desatendiendo, y que no sólo son mejoras posibles que realizar a una nueva aplicación, sino que también generan un nicho de mercado al que se podría apelar.

6 Análisis de requisitos

La especificación de los requisitos permite saber cuáles son las características que debe poseer la aplicación final para considerarse satisfactoria. Para ello, se debe identificar a los principales interesados en el desarrollo de la aplicación, o stakeholders, y sus deseos en cuanto a funcionalidades para la misma.

Antes de proseguir con el detalle de requisitos cabe remarcar que se sigue como método de trabajo la metodología Agile en que, a través de los sprints, se irán realizando breves demostraciones a los stakeholders de las funcionalidades que se hayan agregado. También se tendrán en cuenta 2 definiciones de persona, dos usuarios básicos para los cuales se definirán diferentes historias de usuario. Para la realización del presente proyecto han sido necesarios 8 sprints, con duraciones aproximadas de 2 semanas para cada uno.

En cualquier equipo de trabajo real, el equipo de trabajo de un sprint estaría formado por 5-10 personas como máximo, pero en este caso en concreto, al tratarse de un TFG, el equipo únicamente estaba formado por un desarrollador y un Product Owner, siendo la figura del Product Owner – Scrum Manager realizada por el Profesor Tutor y la función del Team Developer por el alumno de TFG.

6.1 Interesados en el desarrollo (stakeholders)

- Instituto Valenciano de Investigaciones Agrarias (IVIA). Es el principal interesado en este proyecto de ingeniería software, la organización que se espera que haga uso de la aplicación resultante de este proceso. Por tanto, las opiniones de sus investigadores y sus sugerencias o peticiones tendrán especial importancia durante el desarrollo.



Figura 15 IVIA [10]

- Alejandro Rodríguez, profesor de la UPV que también actúa como cotutor de este mismo proyecto, mantiene una relación cercana con el IVIA, por lo que puede considerársele un intermediario, y, en consecuencia, un segundo stakeholder cuyas opiniones deberán ser muy tenidas en cuenta. Dado que es el interesado más cercano, con el que más se va a mantener la comunicación, y que está habituado al uso de aplicaciones similares a la que se busca desarrollar, también es esperable que la mayoría de sugerencias vengan por su parte, por lo que se le considerará el Scrum manager de este proyecto.
- Usuarios de aplicaciones relacionadas con las imágenes hiperespectrales en general. Conviene tener en cuenta que esta aplicación, aun estando desarrollada en base a las solicitudes de miembros del IVIA, debería cubrir, o poder cubrir con pocos cambios, las necesidades de cualquier posible usuario.

Dada esta situación única, el proceso de búsqueda de requisitos más importante serán las conversaciones con empleados del IVIA y con el tutor del proyecto, que sabrán identificar todas aquellas funciones y comodidades que deseen. Aun así, también será importante todo lo que se pueda aprender de 2 procesos más: la definición de personas, y los escenarios de uso. A continuación, se exponen los resultados de ambos procesos:

6.2 Personas

La descripción de una persona es una técnica de elicitación de requisitos que puede resumirse en un ejercicio de empatía por parte de los encargados del proceso, hacia un grupo de posibles usuarios. Consiste en inventar a una persona, con todos los datos que puedan influir en su perspectiva, y tratar de imaginar qué desearía este usuario, para poder así realizar los cambios necesarios para que ese usuario estuviera satisfecho. Se busca una definición precisa en detalles como la edad o sus intereses, pero general en cuanto a sus deseos sobre la aplicación, es decir, no merece el esfuerzo imaginar a un usuario que desea que la forma de un botón concreto de la interfaz sea redonda, pero sí a un usuario que desea que los botones de la interfaz sean vistosos. A continuación, se menciona a dos personas que podrían estar interesadas en el uso de la aplicación, y se trata de discernir qué cualidades considerarían más importantes ambas personas:

Usuario Tipo A

- Trabajo: Investigador especializado en alimentos
- Edad: 35 años

Conocimientos técnicos:

- Escasos conocimientos en cuanto a ordenadores y dispositivos electrónicos.
- Acostumbrado a emplear aplicaciones para obtener información sobre imágenes hiperespectrales.

Objetivos:

- Desea poder utilizar las funciones más básicas de la aplicación sin necesidad de moverse por innecesarios menús que no comprende.
- Desea poder iniciar la aplicación y no tener que preocuparse por especificaciones técnicas.
- Desea poder trasladar el conocimiento obtenido tras años de uso de aplicaciones similares a esta, sin necesidad de ninguna preparación extra.

Usuario Tipo B

- Trabajo: Ingeniero informático
- Edad: 22 años

Conocimientos técnicos:

- Graduado en ingeniería informática. Tiene experiencia con Python.
- No tiene conocimientos técnicos sobre imágenes hiperespectrales, y no sabía de su existencia hasta que se unió a un proyecto relacionado con ellas.

Objetivos:

- Desea poder añadir funciones nuevas al código sin necesidad de manipular demasiadas líneas de código ya presentes, sin necesidad de estudiarlo a un nivel demasiado profundo, y sin perder una cantidad excesiva de tiempo.
- Dispone de código obtenido en su trabajo en otra aplicación sobre imágenes hiperespectrales, y desea poder incluirlo en esta aplicación sin necesidad de realizar muchos cambios, ni en su propio código, ni en el de la aplicación.

6.3 Escenarios de uso

Los escenarios de uso son breves descripciones de un ejemplo de uso de la aplicación, en que se especifican los pasos que seguiría un usuario para obtener un resultado deseado en la aplicación, partiendo desde el estado inicial de la misma. A continuación, se exponen dos escenarios que podrían vivir dos usuarios similares a las dos personas anteriores:

Primer escenario (El usuario no es experto en ordenadores):

El usuario inicia la aplicación y pulsa el botón que le permite seleccionar una imagen y abrirla. Selecciona la imagen concreta que desea en el buscador, y la aplicación tarda unos segundos en procesarla y abrirla. El usuario pulsa el botón para poder seleccionar un área en lugar de un píxel concreto de la imagen, y selecciona la mayor parte del área ocupada por el objeto captado, comprobando que los datos no son los que esperaba. Pulsa en ese momento el botón para realizar la normalización y se le muestra la imagen de referencia blanca, selecciona la región de esta a emplear como referencia, y espera unos segundos a que la aplicación procese nuevamente la imagen. Finalmente, vuelve a seleccionar el área ocupada por el objeto, y comprueba que los datos ofrecidos sí se corresponden con los que esperaba.

Segundo escenario (El usuario es un empleado informático no familiarizado con las imágenes hiperespectrales):

El usuario abre su editor de código y comprueba las variables de que ya dispone la aplicación, así como las librerías ya importadas. Hecho esto, puede comenzar a trabajar en una nueva función que pasará a formar parte de la aplicación. El proceso de familiarización con el código, incluso sin haber leído ningún manual de usuario, es breve, y el usuario no tiene excesivos problemas introduciendo cambios.

6.4 Requisitos

Tras una conversación con el tutor, se pudieron identificar las siguientes funcionalidades, que serían consideradas las más importantes e imprescindibles en una versión inicial de la aplicación. Sin entrar en detalles técnicos o de implementación, serían las siguientes:

- La aplicación debe permitir abrir imágenes hiperespectrales, empleando la información de los ficheros y los headers para mostrarlas en un formato y tamaño correctos.
- La aplicación debe mostrar una imagen de falso color, empleando la información de los tres colores primarios en el espectro electromagnético: rojo, verde y azul.
- La imagen de falso color debe ser modificable por el usuario, que podrá seleccionar las longitudes de onda que se tomará, en cada canal, como referencia para generar la imagen con color falso.
- Seleccionando el usuario un píxel concreto de la imagen hiperespectral, la aplicación debe generar un gráfico que muestre el espectro electromagnético a lo largo de las bandas de la imagen en ese píxel. También debe poder realizarse el mismo gráfico al seleccionar un área rectangular de la imagen, y con un área dibujada totalmente a mano por el usuario.
- La aplicación debería permitir realizar la normalización de la imagen hiperespectral seleccionada. Este es un proceso que permite eliminar imprecisiones e imperfecciones de la imagen y permite tratarla de una forma más apropiada.
- La aplicación debe poder realizar la segmentación de la imagen. Este proceso consiste en separar el objeto de interés (región de interés o ROI) del fondo, permitiendo trabajar solo con el objeto, que se considera prioritario, y eliminar el resto de la imagen.
- La aplicación debe permitir almacenar en un fichero externo multitud de datos de la propia imagen y del espectro electromagnético en la misma, incluyendo la media, mediana, desviación estándar, y algunos percentiles.

7 Situación de desarrollo y mercado

En esta sección se analizarán las ventajas y desventajas que tiene esta aplicación, tanto en relación a su proceso de desarrollo, como en cuanto a su posible desempeño una vez hecha pública:

7.1 Ventajas del proyecto

La situación y stakeholders comentados con anterioridad, ofrecen varias ventajas considerables para el desarrollo, entre ellas:

- Los principales interesados en el desarrollo de la aplicación han tenido acceso a otras aplicaciones del mismo tipo en el pasado, lo que significa que conocen su funcionamiento y las funciones que suelen ofrecerse.
- Dado que los principales interesados son también sus usuarios objetivos, los requisitos de la aplicación no serán basados en conjeturas sobre lo que desean los usuarios, sino en opiniones formadas de personas interesadas en emplear la aplicación.
- Dado el punto anterior, también se dispone de la experiencia y opinión de usuarios expertos en la materia, que podrán solicitar que se prioricen funcionalidades o mejoras que saben que son más importantes.
- Dado que tanto los investigadores del IVIA como Alejandro Rodríguez son personas con cierto conocimiento técnico, las especificaciones técnicas podrán ser discutidas directamente con ellos, y no deberán ser escogidas, evitando así que funcionalidades añadidas posteriormente choquen con el código desarrollado de forma previa.

7.2 Limitaciones

Pese a las ya mencionadas ventajas para el desarrollo, el proyecto también tiene algunas limitaciones que deberán tenerse en cuenta durante su desarrollo para garantizar que ninguno de ellos cause retrasos excesivos o problemas en el transcurso del trabajo. Los problemas detectados, así como la forma en que se espera lidiar con ellos o mitigarlos, serán enumerados a continuación:

- El programador principal no tiene conocimientos previos sobre las imágenes hiperespectrales a nivel teórico ni técnico. Dado que los interesados en el desarrollo de la aplicación son expertos y sí tienen considerable experiencia, y dado un periodo breve de unas pocas semanas en que se buscará información al respecto, cabe esperar que esta solo resulte una limitación en teoría, pero no genere ningún problema real en la práctica.
- A consecuencia del inconveniente anterior, es de esperar que la aplicación no posea pequeñas correcciones que se han aplicado a otras aplicaciones para mejorar su usabilidad, y que serían echadas en falta por usuarios expertos. Se espera contrarrestar esto teniendo constantes revisiones con personas expertas en la materia, que podrían ofrecer feedback e ideas para posibles mejoras, ya sean estas basadas en lo probado en otras aplicaciones, o ideas propias que se les ocurran para mejorar la aplicación.
- El programador del proyecto dispone de un tiempo limitado, debido a que compagina la mayor parte del periodo de desarrollo de esta aplicación, con el desarrollo de sus prácticas en empresa. Los periodos de desarrollo, o sprints, serán relativamente prolongados para permitir unas sesiones de codificación más breves, con lo que se espera disponer de tiempo suficiente para completar las funciones esperadas.

8 Diseño de la solución

En la siguiente sección, se entrará en detalle sobre la forma en que se pretende realizar el trabajo de la aplicación, las herramientas a utilizar, y la forma en que se busca organizar el código.

8.1 Arquitectura del sistema

Dado que se pretende facilitar la modificación e inteligibilidad del código, se optará por un patrón similar al Modelo-Vista-Controlador, en que se dividirá el código en 2 ficheros:

- El primero gestionará la interfaz de usuario, invocará las funciones necesarias, y contendrá las funciones enfocadas en la propia interfaz, como cambios de imágenes y trabajo sobre las mismas.
- El segundo fichero contendrá el código de la mayoría de funciones técnicas a emplear para realizar cálculos y cambios ajenos a la propia interfaz de usuario.

Dado que una parte importante del atractivo de la aplicación pasa por permitir a otros usuarios modificarla, se buscará que las funciones estén correctamente documentadas para ser entendidas por personas ajenas a su programación original en la medida de lo posible.

8.2 Materiales

IDE

Se empleó Visual Studio Code como editor de código principalmente por la comodidad que ofrece, además de ser un sistema al que el programador ya está acostumbrado.

Visual Studio Code es un entorno de desarrollo publicado en 1997, que sigue recibiendo soporte y que es compatible con un buen número de lenguajes de programación, incluyendo Python, por lo que fue la opción ideal para este desarrollo. [11][12]

Lenguaje de programación

El lenguaje de programación a utilizar será Python. Python es un lenguaje de programación cuya primera versión se liberó al público en 1991. Creado por Guido van Rossum, Python es un lenguaje interpretado y de alto nivel multiparadigma, que soporta orientación a objetos, programación imperativa, y en cierta medida, funcional, que destaca por tener un tipado dinámico. Los motivos para emplearlo son varios: si bien no ha sido el más usado, sí se ha empleado en varias ocasiones a lo largo del grado, y es un lenguaje sencillo de aprender. En segundo lugar, y más importante, se ofrecieron como ayuda algunas funciones en Python relacionadas con el tema de este proyecto, que podrían servir en el futuro para no tener que escribir todo el código desde cero. [13][14]

Librerías empleadas

Una vez seleccionados el lenguaje y la herramienta con que se va a trabajar, es interesante mencionar las librerías a las que se ha dado uso para la consecución de este proyecto. Conviene aclarar, sin embargo, que la aplicación puede ampliarse o modificarse en el futuro, por lo que esta es la sección más susceptible de quedar obsoleta en el futuro.

Las librerías Python empleadas son las siguientes:

- **os:** Permite realizar funcionalidades relacionadas con el sistema operativo, incluyendo abrir ficheros o, en el caso de esta aplicación, acceder a la ruta asociada a un fichero seleccionado. [15]
- **rasterio:** Rasterio es una librería que permite trabajar con imágenes hiperespectrales de diversos formatos.[16]
- **matplotlib:** Matplotlib es una librería centrada en la creación de gráficos en dos dimensiones, y que se emplea en múltiples instancias en esta aplicación. Se emplean los siguientes módulos: [17]
 - **matplotlib.pyplot:** Sirve para definir y dibujar gráficos lineales. [17]
 - **RectangleSelector** de **matplotlib.widgets:** Sirve para dibujar rectángulos entre el área de un click, y el área en que se suelta el ratón. [18]
 - **FigureCanvasQTAgg** de **matplotlib.backends.backend_qt5agg:** Sirve para integrar gráficos en interfaces creadas con PyQt5 [19]
- **numpy:** Es una librería que permite realizar multitud de cálculos y operaciones en Python. [20]
- **PyQt5:** Ofrece varios módulos enfocados en las interfaces de usuario [21]
 - **QtWidgets:** Ofrece múltiples facilidades para la construcción de interfaces de usuario [22]

- **QFileDialog** de **PyQt5.QtWidgets**: Ofrece al usuario una interfaz sencilla para seleccionar un fichero [23]
- **QMessageBox** de **PyQt5.QtWidgets**: Ofrece una interfaz con la que entregar información al usuario sobre el estado de la aplicación [24]
- **Skimage**: Útil para procesar imágenes [25]
 - **skimage.filters**: Permite emplear múltiples filtros en las imágenes [26]
 - **morphology**: Sirve para trabajar con diversas formas en las propias imágenes [27]
- **scipy.ndimage**: Ofrece varias funciones para trabajar con imágenes multidimensionales [28]
- **cv2**: Proporciona una interfaz sencilla para el procesamiento de imágenes [29]
- **time**: Proporciona funciones relacionadas con el tiempo [30]

9 Proceso de desarrollo

La metodología empleada para el desarrollo será Agile, para permitir realizar cambios y modificaciones con rapidez si los usuarios principales lo solicitan, y para obtener un feedback constante por parte de los principales interesados, el proceso de desarrollo se dividirá en sprints, breves periodos de desarrollo en que el esfuerzo se centrará en un limitado número de funciones o mejoras a implementar.

En cada sprint se desarrollará una versión funcional, aunque no necesariamente final o libre de errores, de alguna función de la aplicación. Una vez finalizado ese sprint, se realizará una prueba para el cliente, que servirá para identificar si está satisfecho con el resultado, y tanto en caso de estarlo como de no estarlo, indicará los cambios que desea ver realizados, así como las funciones que desearía ver añadidas a continuación, hayan sido estas comunicadas en las reuniones iniciales del proceso de desarrollo o no.

La comunicación constante con los stakeholders es una de las principales bazas que permitirá que el proyecto salga adelante en un estado que resulte satisfactorio. Usualmente un sprint en este proceso de desarrollo no se alargará más de dos semanas; sin embargo, es posible que alguna función concreta requiera periodos más prolongados para llegar a un estado que se pueda considerar aceptable.

A continuación, se hará un repaso breve de los añadidos o mejoras introducidas en cada sprint, así como del funcionamiento general de la aplicación en cada proceso:

9.1 Sprint 1

Una vez realizado el proceso de documentación básico, se procedió a realizar una versión de la funcionalidad más básica de la aplicación: la posibilidad de abrir y mostrar imágenes hiperespectrales.

Name	Date modified	Type	Size
20240411_MAD_AGU_T0_S1_0001_headw...	11/04/2024 12:58	BIL File	27.198 KB
20240411_MAD_AGU_T0_S1_0001_headw...	11/04/2024 12:58	HDR File	3 KB
20240411_MAD_AGU_T0_S1_0001_ids.bil	11/04/2024 12:58	BIL File	442.423 KB
20240411_MAD_AGU_T0_S1_0001_ids.hdr	11/04/2024 12:58	HDR File	2 KB
20240411_MAD_AGU_T0_S1_0002_headw...	11/04/2024 12:58	BIL File	30.853 KB
20240411_MAD_AGU_T0_S1_0002_headw...	11/08/2024 14:51	TIF File	123.414 KB
20240411_MAD_AGU_T0_S1_0002_headw...	11/04/2024 12:58	HDR File	3 KB
20240411_MAD_AGU_T0_S1_0002_ids.bil	11/04/2024 12:58	BIL File	501.749 KB
20240411_MAD_AGU_T0_S1_0002_ids.hdr	11/04/2024 12:58	HDR File	2 KB
20240411_MAD_AGU_T0_S1_0003_headw...	11/04/2024 12:59	BIL File	26.123 KB
20240411_MAD_AGU_T0_S1_0003_headw...	11/04/2024 12:59	HDR File	3 KB
20240411_MAD_AGU_T0_S1_0003_ids.bil	11/04/2024 12:59	BIL File	424.587 KB
20240411_MAD_AGU_T0_S1_0003_ids.hdr	11/04/2024 12:59	HDR File	2 KB
20240411_MAD_AGU_T0_S1_0004_headw...	11/04/2024 12:59	BIL File	25.585 KB
20240411_MAD_AGU_T0_S1_0004_headw...	11/04/2024 12:59	HDR File	3 KB
20240411_MAD_AGU_T0_S1_0004_ids.bil	11/04/2024 12:59	BIL File	416.444 KB

Figura 16 Captura de la selección de imagen de la aplicación

Tan pronto como se inicia la aplicación, se abre el menú de selección de imágenes hiperespectrales. Pese a que el menú muestra tanto los propios ficheros de imagen como los header, o archivo de cabecera con la información de los metadatos de la imagen, la apertura solo funciona de forma correcta si se selecciona el fichero, y una vez realizada la selección, la única forma de seleccionar otra imagen es cerrar y abrir nuevamente la aplicación. Estas son mejoras que se implementarán en un sprint futuro, pero es importante tenerlas en cuenta lo antes posible.

A continuación, se puede ver el aspecto de la aplicación una vez ha sido seleccionada una imagen:

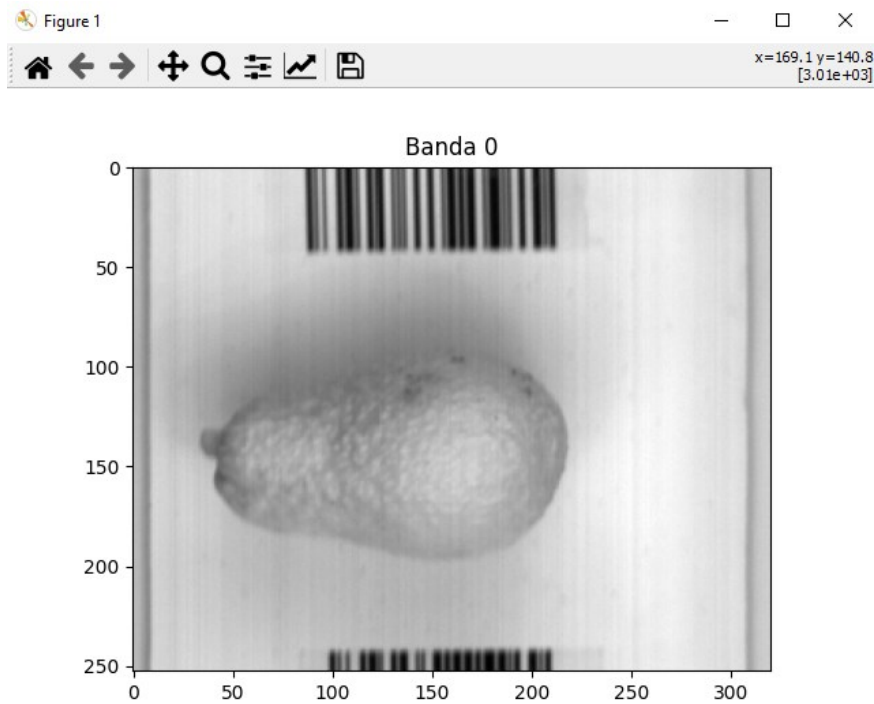


Figura 17 Captura de la primera versión de la aplicación

Como se puede apreciar en la Figura 16, esta versión de la aplicación tiene una interfaz muy básica, que muestra solo la propia imagen, el número de banda (que en esta versión aún no puede ser cambiado por el usuario) y la ubicación en los ejes x e y del cursor.

Las funcionalidades esperadas estaban presentes, pero de forma obvia, había aún muchos cambios a realizar.

En base a este sprint, el Scrum manager solicitó tener la posibilidad de cambiar libremente de banda, funcionalidad en que ya se había comenzado a trabajar.

9.2 Sprint 2

Durante este sprint, el esfuerzo de desarrollo se centró en ofrecer una funcionalidad en que ya se había comenzado a trabajar en sprint anterior.

El usuario puede ahora seleccionar la banda concreta de la imagen que desea visualizar.

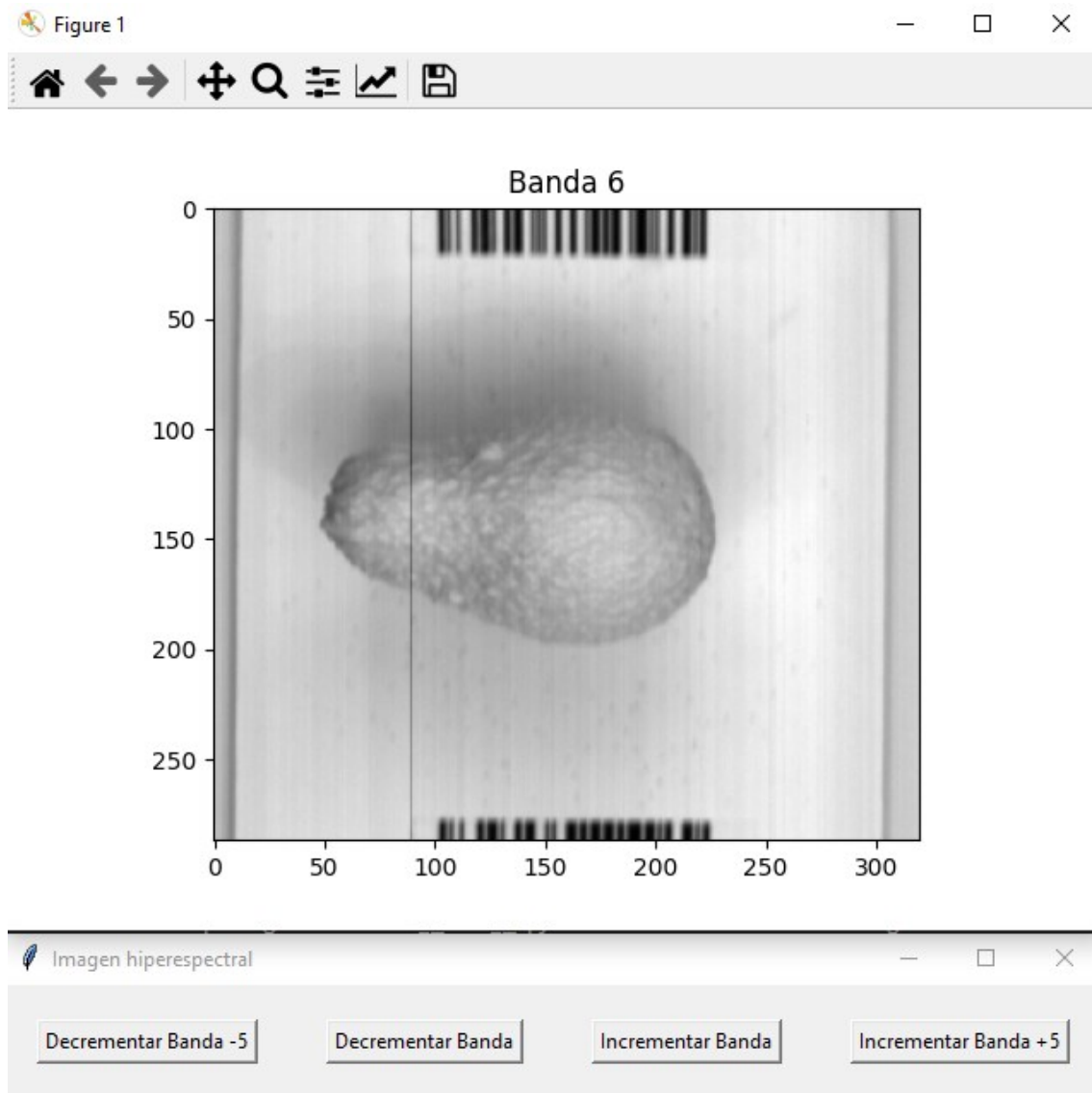


Figura 18 Captura de la segunda versión de la aplicación

Como puede observarse en la Figura 17, el selector de bandas se encuentra en una ventana nueva de la aplicación, por lo que no es ideal ni práctico.

Para el siguiente sprint, se desea comprimir ambas ventanas en una sola. Por otro lado, el Scrum manager también solicitó que se pudieran generar gráficos sobre el espectro electromagnético en base a la selección de un píxel.

9.3 Sprint 3

En el tercer script el esfuerzo se ha centrado en añadir dos funcionalidades importantes:

1. La posibilidad de seleccionar la banda concreta que se va a mostrar desde una misma ventana. El salto puede ser de una o cinco bandas, como en el sprint previo.

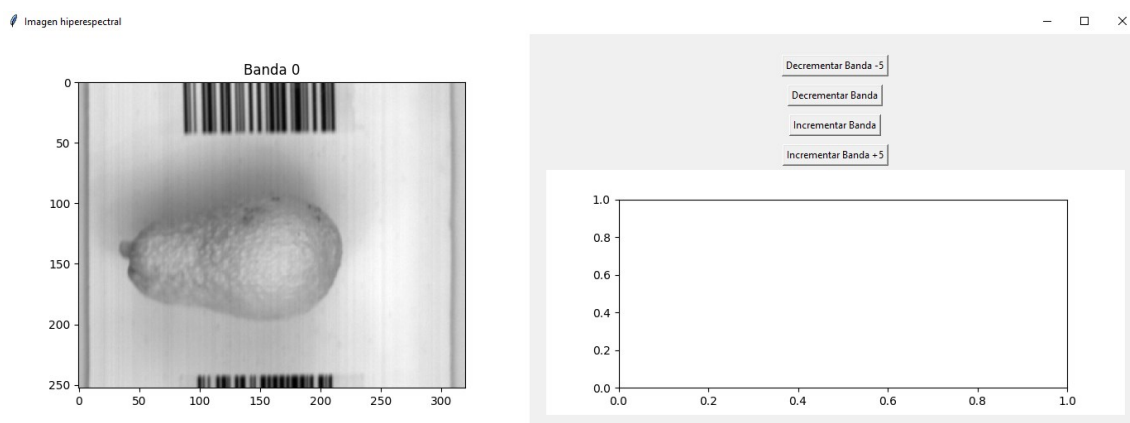


Figura 19 Captura de la interfaz de la tercera versión de la aplicación, en la banda 0

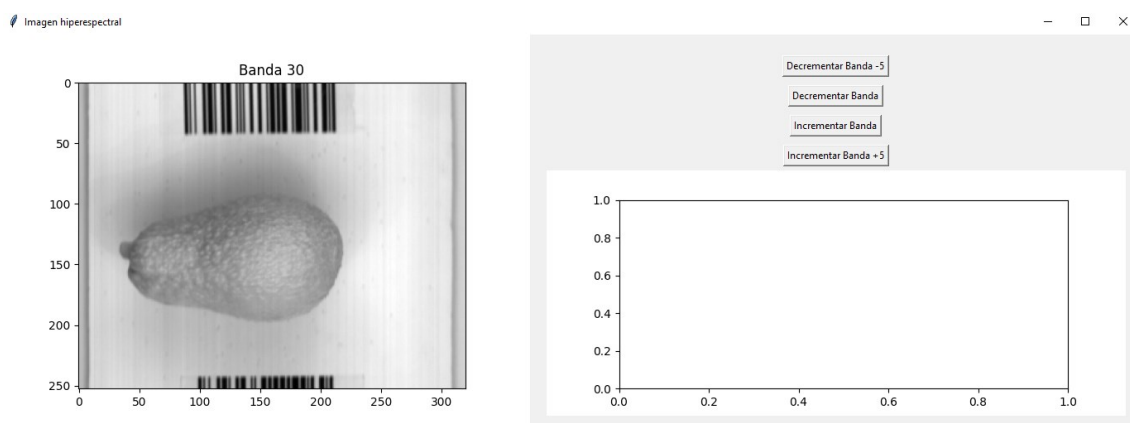


Figura 20 Captura de la interfaz de la tercera versión de la aplicación, en la banda 30

2. La posibilidad de seleccionar un píxel concreto de la imagen, y ver un gráfico en que se representa el espectro electromagnético en ese píxel a través de todas las bandas de la imagen.

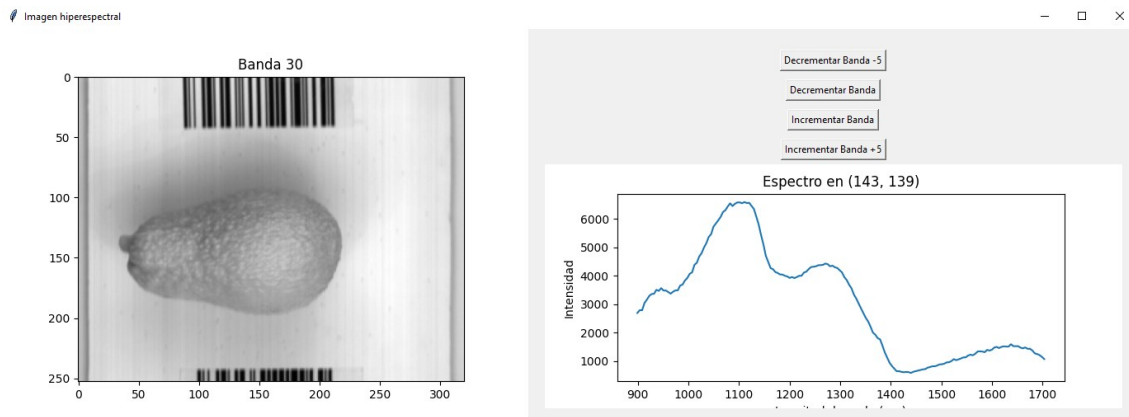


Figura 21 Captura de la aplicación mostrando el gráfico generado al seleccionar un punto de la imagen

Las funcionalidades solicitadas por el Scrum manager tras el anterior sprint estaban presentes, y en un estado que le resultó satisfactorio. Sin embargo, también solicitó que fuera añadida una nueva funcionalidad, consistente en generar con tres bandas o canales seleccionados de cualquier longitud de onda, una misma imagen a color.

9.4 Sprint 4

En el cuarto script, el esfuerzo de trabajo se centró en desarrollar la función solicitada en la revisión del script anterior, concretamente:

1. Generar una imagen de falso color, basada en la selección de tres canales o bandas disponibles en el hipercubo. Esta creación debe estar disponible tan pronto como se abra la imagen original, y no se requiere ninguna interacción con ella. También solicitó que, en el propio gráfico, se mostrasen líneas que representasen los valores que se seleccionaron para las bandas espectrales de los tres colores primarios (rojo, verde y azul), cambio que se pudo implementar poco después de la demostración.

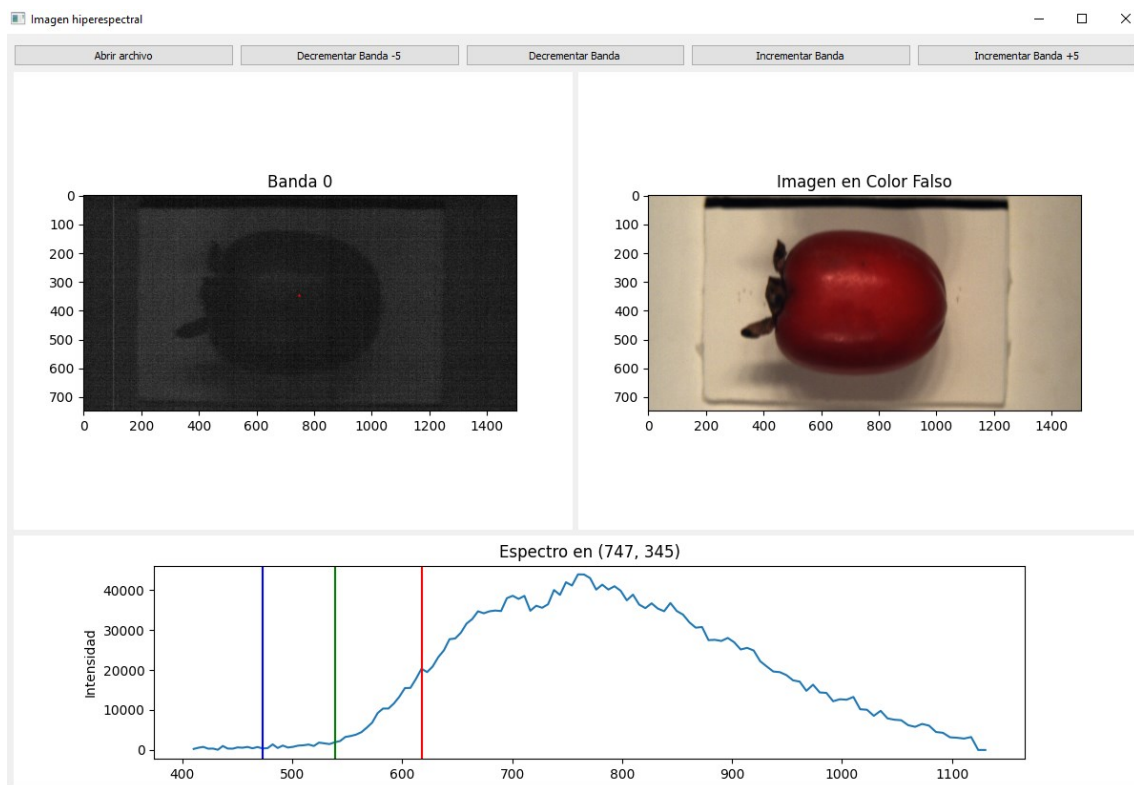


Figura 22 Captura de la aplicación que muestra la funcionalidad de falso color

2. También se ofrece en esta versión la posibilidad de cambiar la imagen seleccionada una vez iniciada la aplicación y seleccionada una imagen hiperespectral, con la pulsación de un botón. Sin embargo, sigue existiendo la limitación previa que obliga a seleccionar el fichero, e impide seleccionar el header de la propia imagen, pese a que aparece en las opciones de la búsqueda.

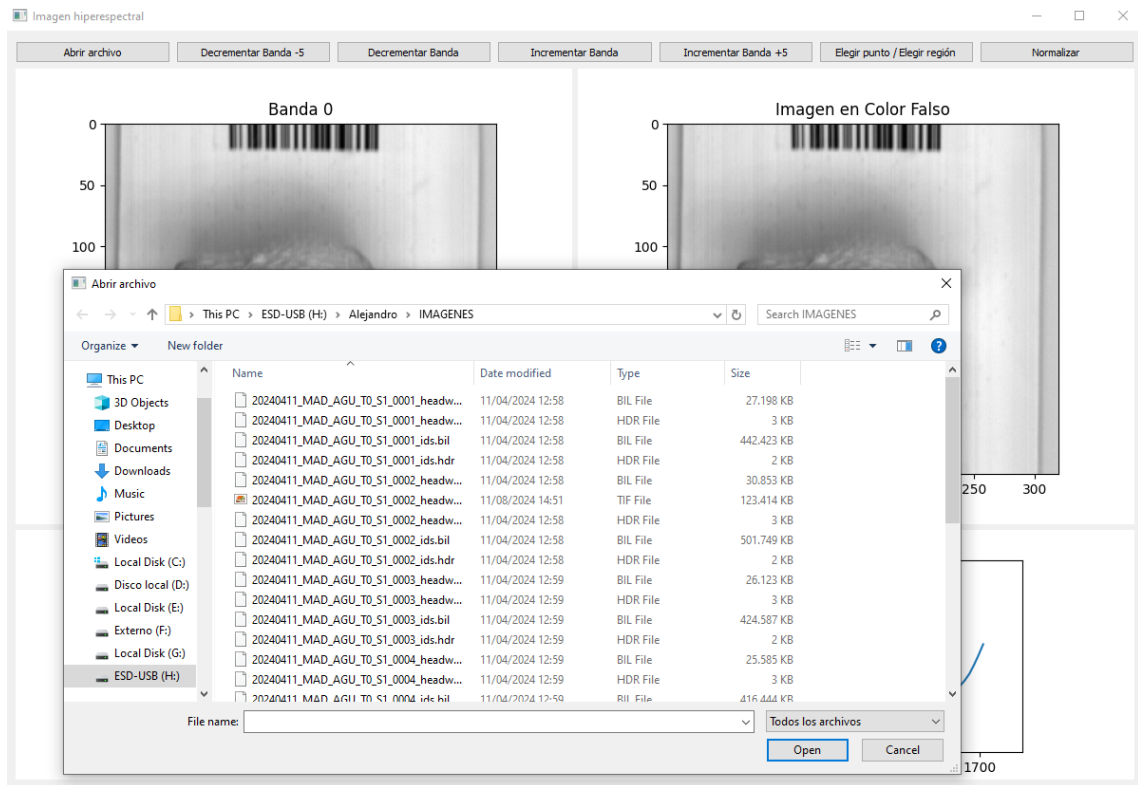


Figura 23 Captura de la selección de imagen de la aplicación

Debido a las cualidades de las propias imágenes, no todas tienen la posibilidad de ofrecer una imagen a color convincente. Sin embargo, el Scrum manager se mostró satisfecho con esta funcionalidad igualmente, y solicitó añadir otra: la posibilidad de seleccionar un área rectangular de la imagen para la representación del espectro medio de la región seleccionada, en lugar de la limitada selección del espectro de un solo punto o píxel.

9.5 Sprint 5

Durante este sprint, se desarrolló la funcionalidad solicitada al final del sprint anterior, consistente en:

1. Añadir la posibilidad de seleccionar un área rectangular (o región de interés rectangular) en lugar de un píxel concreto para la representar el espectro medio dentro de esa región. Se muestran igualmente los puntos entre los que se encuentra esta región.

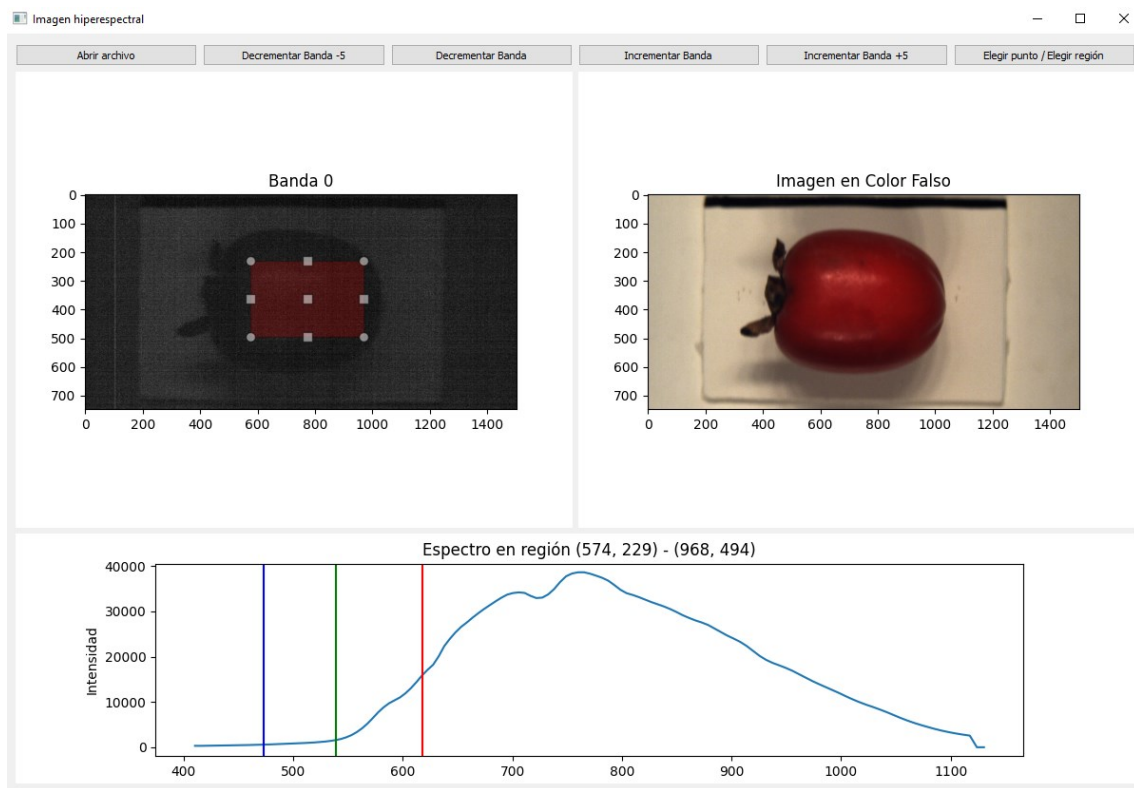


Figura 24 Captura de la funcionalidad de selección de un área rectangular de la aplicación

2. Se puede desplazar el rectángulo simplemente al seleccionarlo y automáticamente se cambia el espectro representado.

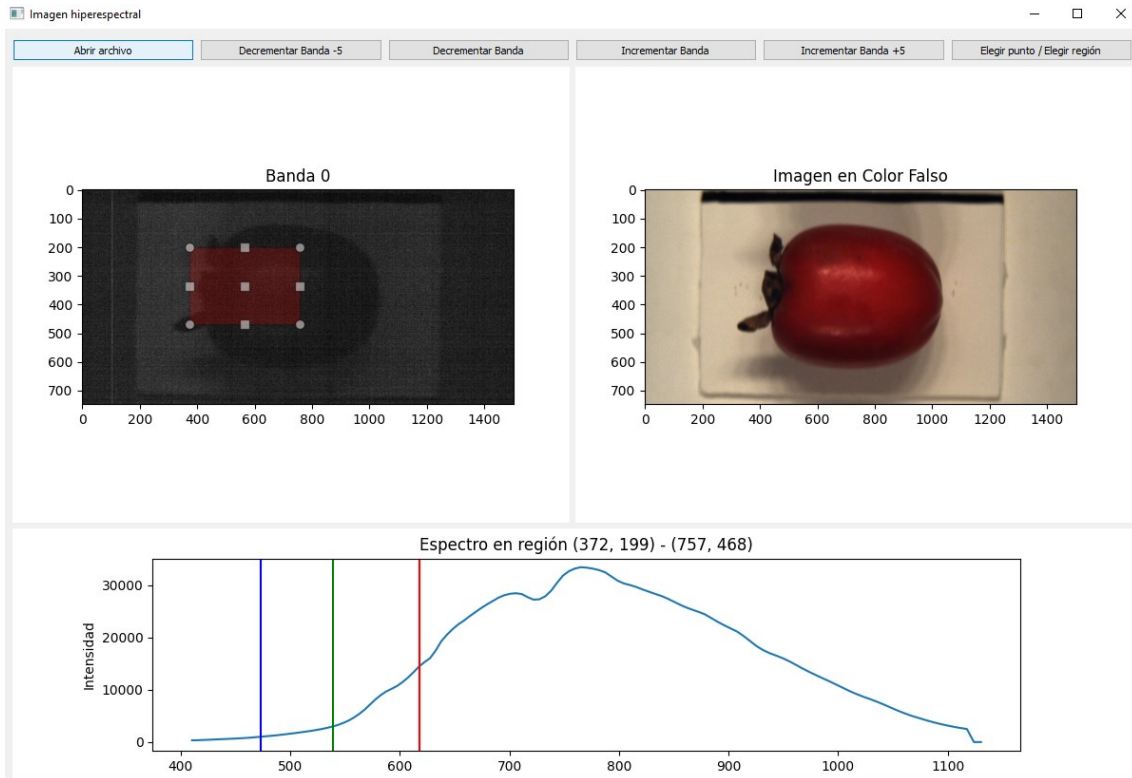


Figura 25 Captura de la funcionalidad de selección de un área rectangular de la aplicación

3. Se puede modificar el tamaño del rectángulo sin necesidad de dibujar un nuevo rectángulo.

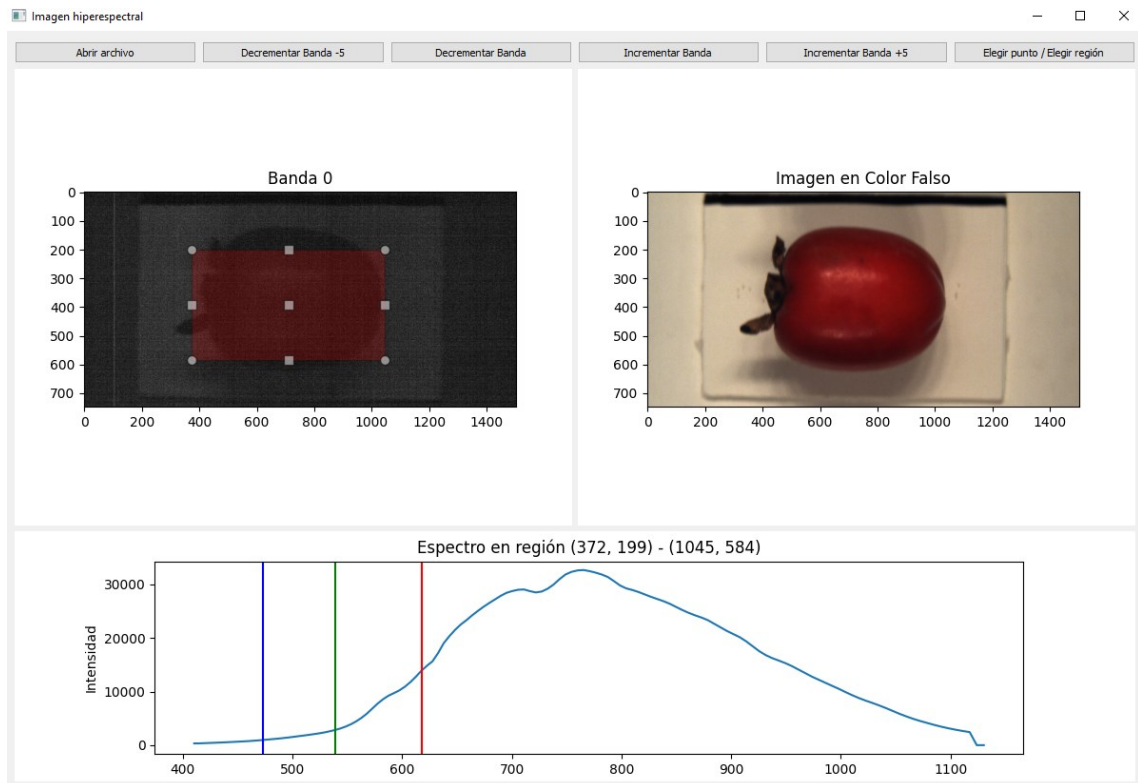


Figura 26 Captura de la funcionalidad de selección de un área rectangular de la aplicación

El Scrum manager se mostró satisfecho con esta función, y solicitó un nuevo cambio en relación a la simulación de imagen a color: de base, se seguirá mostrando la imagen a color como se había hecho hasta el momento, pero también se podrá seleccionar manualmente la longitud de onda que se desea emplear como referencia para cada canal, en lugar de utilizar las longitudes de onda del rojo, verde y azul.

9.6 Sprint 6

En este sprint, se implementaron las siguientes mejoras:

- El usuario a partir de este punto puede seleccionar las longitudes de onda que tomar para generar la imagen de falso color. En lugar de obligar al usuario a introducir valores a seleccionar, se le da la opción de seleccionar y arrastrar las líneas que representan cada uno de los colores primarios, para escoger la longitud de onda que se empleará en el proceso de coloreado.

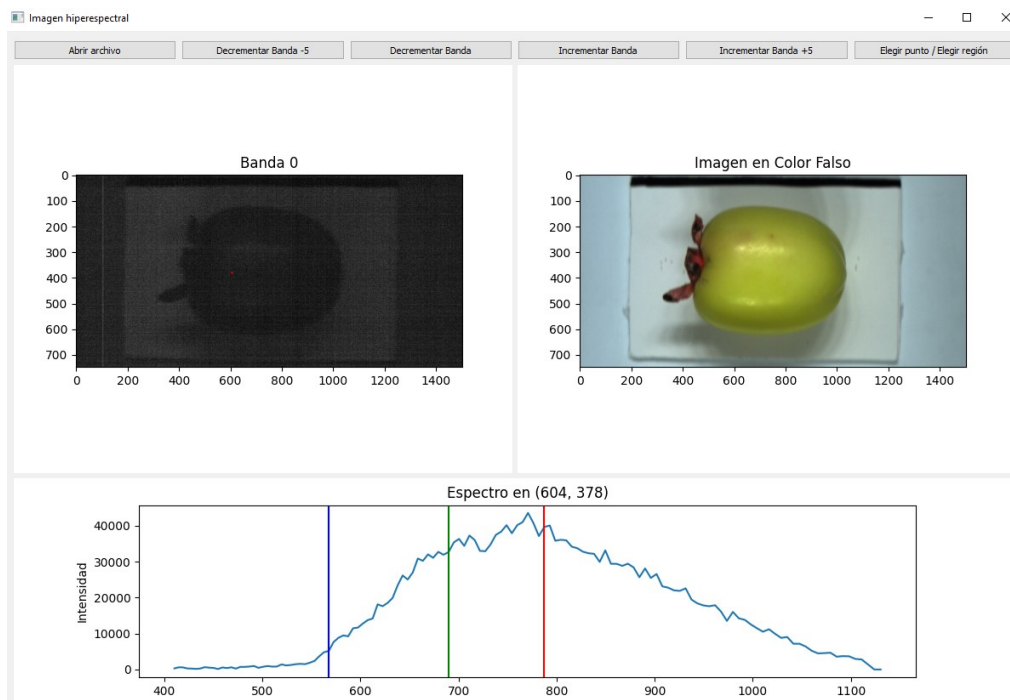


Figura 27 Captura del resultado de la selección de longitudes de onda de la aplicación

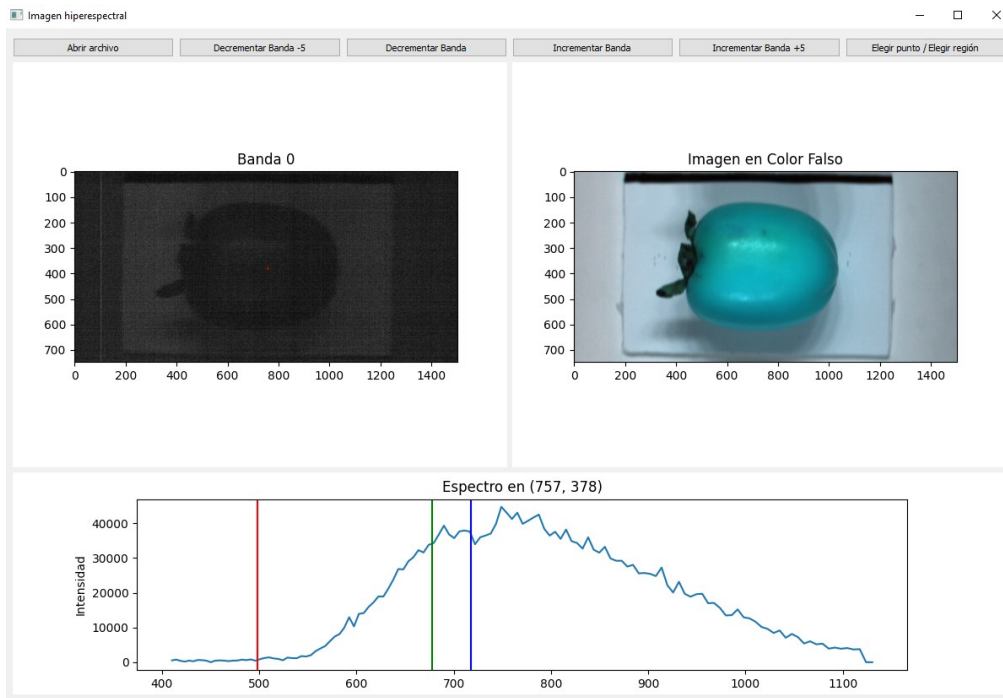


Figura 28 Captura del resultado de la selección de longitudes de onda de la aplicación

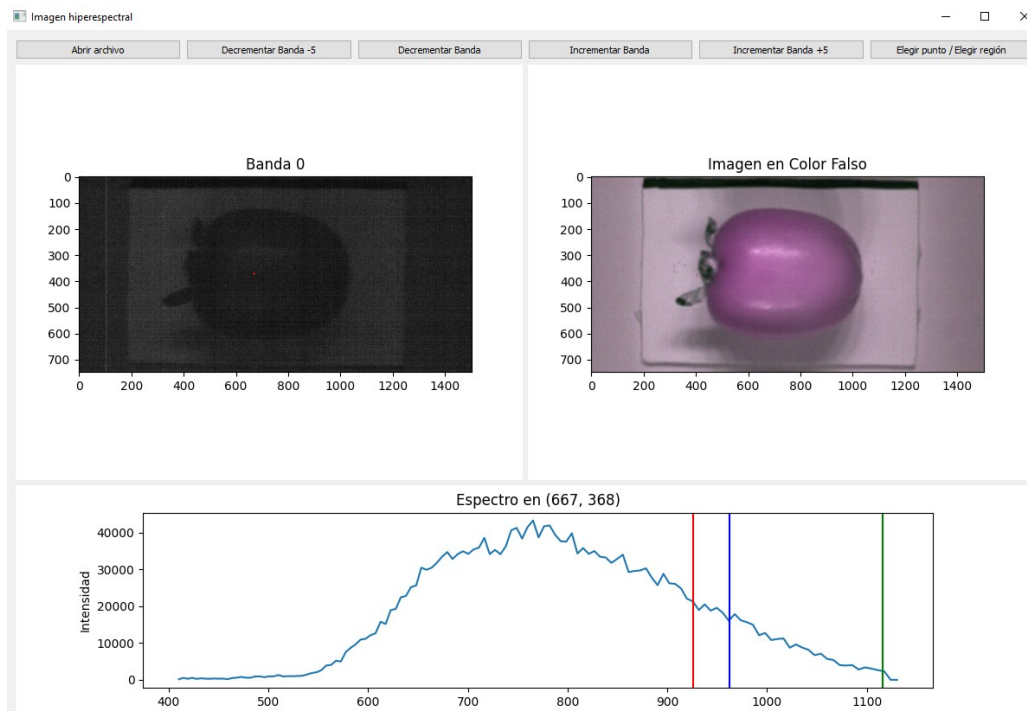


Figura 29 Captura del resultado de la selección de longitudes de onda de la aplicación

- Además, se mejoró el código de selección de imágenes, para permitir seleccionar tanto el propio fichero de la imagen, como su header. Del mismo modo, ahora se permite al usuario distinguir durante la búsqueda para que se le muestren todos los ficheros, o únicamente los headers.

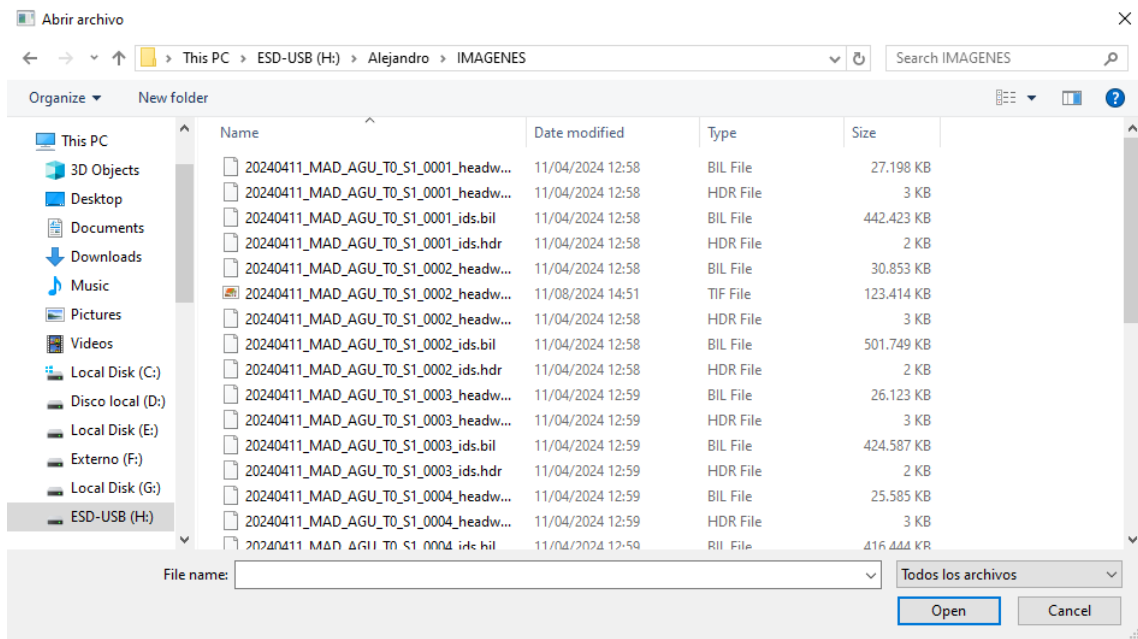


Figura 30 Captura de la búsqueda de imágenes de la aplicación

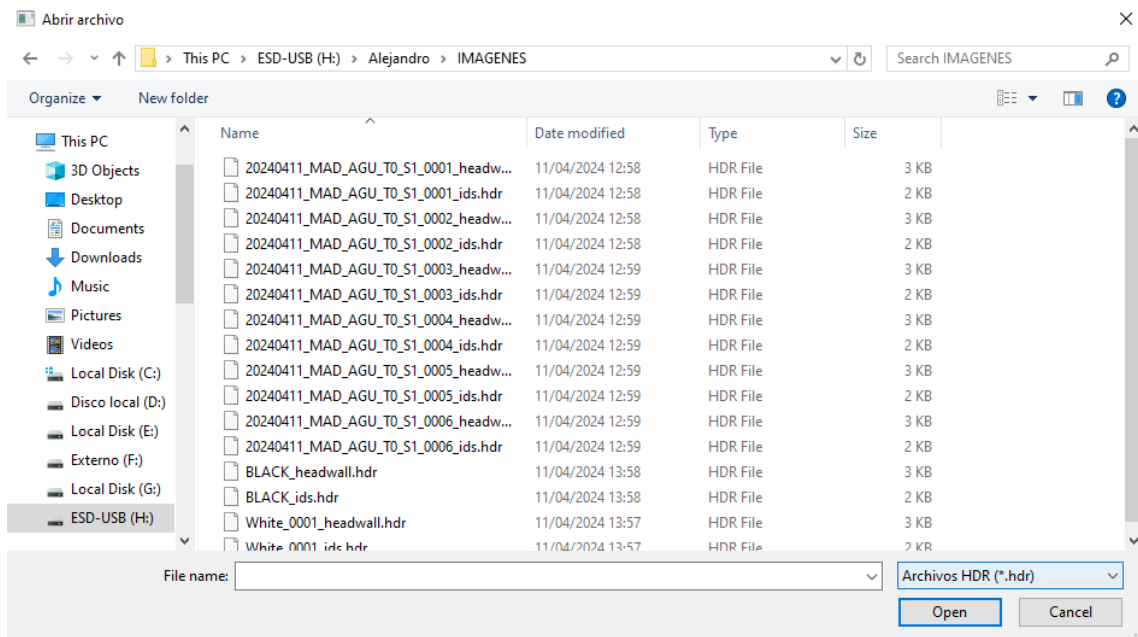


Figura 31 Captura de la búsqueda de imágenes de la aplicación

El Scrum manager se mostró satisfecho con el resultado de este sprint, y solicitó que para el sprint siguiente se introdujeran dos funciones pertenecientes a una aplicación previa, desarrollada por él mismo.

9.7 Sprint 7

Para este sprint, el Scrum manager había solicitado que se introdujeran las dos funciones siguientes en la aplicación:

- Función de `segmentación_otsu.py` (Figura 31): Esta función permite utilizando el método de Otsu o método del valor umbral [31] es una técnica utilizada en el procesamiento de imágenes para realizar una segmentación automática de una imagen en dos clases, región de interés (u objeto) y fondo. Esta método es útil para clasificar los pixeles de una imagen en dos grupos basados su intensidad, sin necesidad de especificar un umbral de forma manual.

```
9
10 import skimage.filters
11 from skimage import morphology
12 import scipy.ndimage
13 import numpy as np
14
15 def segmentacion_otsu(imagen):
16
17     median_filtered = scipy.ndimage.median_filter(imagen, size=3) # Aplicamos un filtro de mediana para mejorar la segmentacion
18
19     ##result = skimage.filters.thresholding.try_all_threshold(median_filtered) # ver todos los metodos de umbralizacion
20     thr=skimage.filters.threshold_otsu(median_filtered)
21
22     mask=np.uint8(median_filtered>thr)*255 # devuelvo la mascara como uint8
23
24     #Eliminamos posibles objetos pequeños que hayan quedado
25     morphology.remove_small_objects(mask, 20, in_place=True)
26     # Rellenamos posibles huecos que hayan quedado
27     seed = np.copy(mask)
28     seed[1:-1, 1:-1] = mask.max()
29
30     mask = morphology.reconstruction(seed, mask, method='erosion')
31
32     return mask.astype(np.uint8)
33
```

Figura 32 Captura de la primera función a incluir

- Función `superposition_mask.py` (Figura 32): Esta función permite superponer la máscara (imagen binaria) obtenida de la función `segmentación_otsu.py` con la imagen a segmentar, guardando el resultado como un thumbnail.

```

-
9  import cv2
10 import numpy as np
11
12 def superposition_mask(image, mask, folder_save):
13
14     # Normalizacion de la imagen para convertirlo UINT8 0-255
15     image_u8=cv2.normalize(image,None, 0,255,cv2.NORM_MINMAX,cv2.CV_8U)
16     image_u8=cv2.cvtColor(image_u8,cv2.COLOR_GRAY2BGR)
17
18     redImg = np.zeros((image_u8.shape[0],image_u8.shape[1],3), image_u8.dtype)
19     redImg[:,:] = (0, 0, 255)
20     redMask = cv2.bitwise_and(redImg, redImg, mask=mask)
21     cv2.addWeighted(redMask, 0.6, image_u8, 1, 0, image_u8)
22
23     #cv2.imshow('image_u8',image_u8)
24
25     # Guardamos un thumblair
26     size=(image_u8.shape[1]//2,image_u8.shape[0]//2)
27     imRes = cv2.resize(image_u8,size,interpolation=cv2.INTER_AREA)
28     #cv2.imshow('imRes',imRes)
29     cv2.imwrite(folder_save, imRes)
30     #im_super = cv2.bitwise_and(imageColor, mask_aux)

```

Figura 33 Captura de la segunda función a incluir

El objetivo principal de este sprint al introducir estas dos funciones no era solo el de añadir nuevas funcionalidades a la aplicación, sino el de demostrar que se pueden introducir funciones aisladas con facilidad, sin necesidad de realizar cambios excesivos al resto de la aplicación, y haciéndolas funcionar.

La versión final de ambas funciones, tras realizar un número reducido de cambios debidos a cambios en las librerías utilizadas, y otros destinados a generar imágenes visibles para confirmar que los resultados son correctos, es la siguiente:

```

def segmentacion_otsu(imagen):
    median_filtered = scipy.ndimage.median_filter(imagen, size=3) # Aplicamos un filtro de mediana para mejorar la segmentacion
    ##result = skimage.filters.thresholding.try_all_threshold(median_filtered) # ver todos los metodos de umbralizacion
    thr=skimage.filters.threshold_otsu(median_filtered)

    mask=np.uint8(median_filtered<thr)*255 # devuelvo la mascara como uint8
    cv2.imshow('mask', mask)
    #Eliminamos posibles objetos pequeños que hayan quedado
    morphology.remove_small_objects(mask, 20)
    # Rellenamos posibles huecos que hayan quedado
    seed = np.copy(mask)
    seed[1:-1, 1:-1] = mask.max(0)

    mask = morphology.reconstruction(seed, mask, method='erosion')

    return mask.astype(np.uint8)

def superposition_mask(image, mask, folder_save):
    # Normalizacion de la imagen para convertirlo UINT8 0-255
    image_u8=cv2.normalize(image,None, 0,255,cv2.NORM_MINMAX,cv2.CV_8U)
    image_u8=cv2.cvtColor(image_u8,cv2.COLOR_GRAY2BGR)

    redImg = np.zeros((image_u8.shape[0],image_u8.shape[1],3), image_u8.dtype)
    redImg[:,:] = (0, 0, 255)
    redMask = cv2.bitwise_and(redImg, redImg, mask=mask)
    imagen_superpuesta = cv2.addWeighted(redMask, 0.6, image_u8, 1, 0, image_u8)

    #cv2.imshow('image_u8',image_u8)
    cv2.imshow('imagen superpuesta', imagen_superpuesta)
    cv2.imshow('redMask', redMask)
    # Guardamos un thumblair
    """size=(image_u8.shape[1]//2,image_u8.shape[0]//2)
    imRes = cv2.resize(image_u8,size,interpolation=cv2.INTER_AREA)
    cv2.imshow('imRes',imRes)"""
    ##cv2.imwrite(folder_save, imRes)
    #im_super = cv2.bitwise_and(imageColor, mask_aux)

```

Figura 34 Captura del código de la aplicación

La introducción de estas dos nuevas funciones no genera ningún problema, ni costó demasiado tiempo hacer que fuesen correctamente invocadas. Sin embargo, como se puede ver en la Figura 34, hubo ligeros problemas con los resultados.

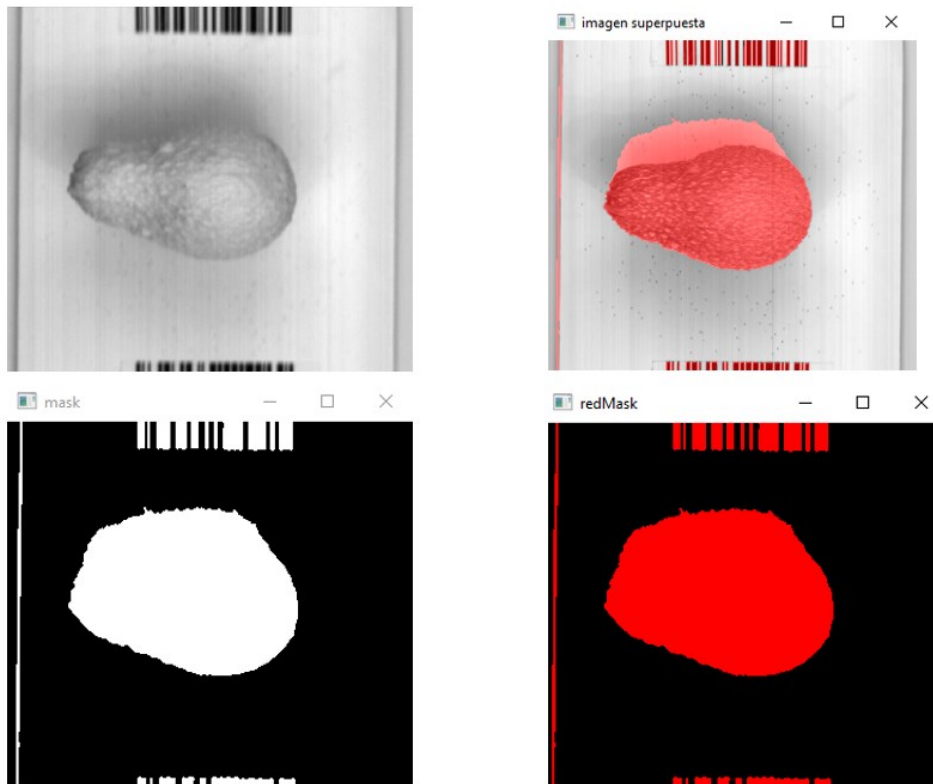


Figura 35 Imágenes resultado de las funciones introducidas

Como se puede apreciar, las funciones trabajan perfectamente, pero las imágenes ofrecidas para las pruebas tienen alguna tara que dificulta su correcto funcionamiento. En las capturas se puede apreciar una sombra, debido a un fallo en la iluminación durante la adquisición, que dificulta al algoritmo diferenciar entre el objeto y el fondo, haciendo que la superposición falle.

Sin embargo, el principal objetivo de este sprint era comprobar cómo se podía introducir una nueva función ajena al código, y el proceso ha sido tan sencillo como introducir las nuevas funciones, e introducir las invocaciones donde fueran necesarias, por lo que puede considerarse un sprint exitoso, si bien será deseable realizar pruebas con imágenes de mayor calidad en el futuro.

Tras la revisión de este script, el Scrum manager solicitó que se introdujera una nueva función que llevase a cabo la calibración de la imagen hiperespectral, y ofreció una breve explicación de en qué consiste este proceso, que será expuesta en el resumen del sprint siguiente.

9.8 Sprint 8

Tras el sprint previo, el Scrum manager solicitó que se introdujera una función que realizase la calibración de la imagen hiperespectral.

La calibración es un proceso destinado a ajustar los datos crudos capturados por el sensor de la cámara hiperespectral para representar con precisión la reflectancia de la escena observada. La calibración busca corregir las distorsiones y variaciones que pueden estar presentes debido a las características del sensor, las condiciones ambientales y otros factores externos asegurando que los valores espectrales obtenidos sean fiables y comparables con otras mediciones realizadas en otros días y otros sistemas de adquisición.

Uno de los métodos más comunes para realizar la calibración radiométrica es el uso de referencias blancas y negras, las cuales son superficies con propiedades reflectivas bien conocidas y controladas.

La referencia blanca es una superficie o material que posee una alta reflectancia de casi el 100% de la luz incidente en todas las longitudes de onda relevantes para la imagen hiperespectral.

En cambio, la referencia negra es una superficie con una reflectancia muy baja diseñada para absorber casi toda la luz incidente. Es utilizada para medir y corregir el ruido de fondo y la respuesta del sensor a la ausencia de luz.

El proceso de calibración empieza durante el proceso de adquisición de imágenes, donde:

- 1.- Previo a la adquisición de cualquier muestra bajo estudio, se captura una imagen de referencia blanca bajo las mismas condiciones de iluminación y con el mismo sensor que se va a utilizar en el estudio.
- 2.- A continuación, se procede con una adquisición de la referencia negra también bajo las mismas condiciones de adquisición que la referencia blanca.
- 3.- Por último, se adquieren las imágenes de las muestras bajo estudio.

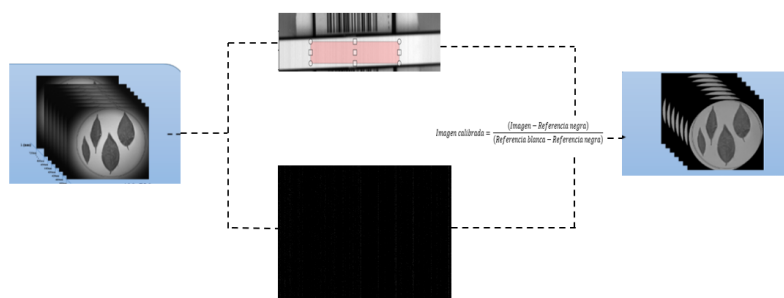


Figura 36 Muestra visual del proceso de normalización

Una vez adquirido todo, el proceso de calibración pasa por restar la referencia negra a la imagen hiperespectral, y dividir la imagen resultante entre la imagen obtenida al restarle a la referencia blanca la negra. En concreto, el proceso se ve en la siguiente ecuación:

$$\text{Imagen calibrada} = \frac{(\text{Imagen} - \text{Referencia negra})}{(\text{Referencia blanca} - \text{Referencia negra})} \quad (3)$$

Ecuación que dará como resultado un hipercubo con valores de intensidad entre 0 y 1.

Sin embargo, debe tenerse en cuenta un factor extra, y es que la imagen de referencia blanca, al contrario que la imagen del objeto bajo estudio, no tiene la misma resolución espacial y ocupa un espacio reducido en ella. Por tanto, se debe seleccionar, previo a la calibración, la región de interés que ocupa la referencia blanca, y calcular el espectro de referencia blanca que se utilizará en la ecuación anterior.

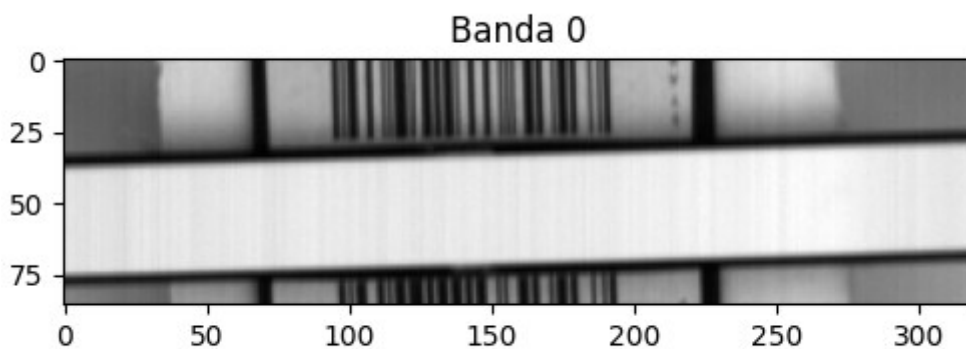


Figura 37 Banda 0 de la imagen de referencia blanca

La intención inicial pasaba por seleccionar a través del código la región de la imagen que se usará como referencia, pero tras consultarlo con el usuario, se optó por hacer que sea el propio usuario el que deba seleccionar el área de la imagen a usar como referencia.

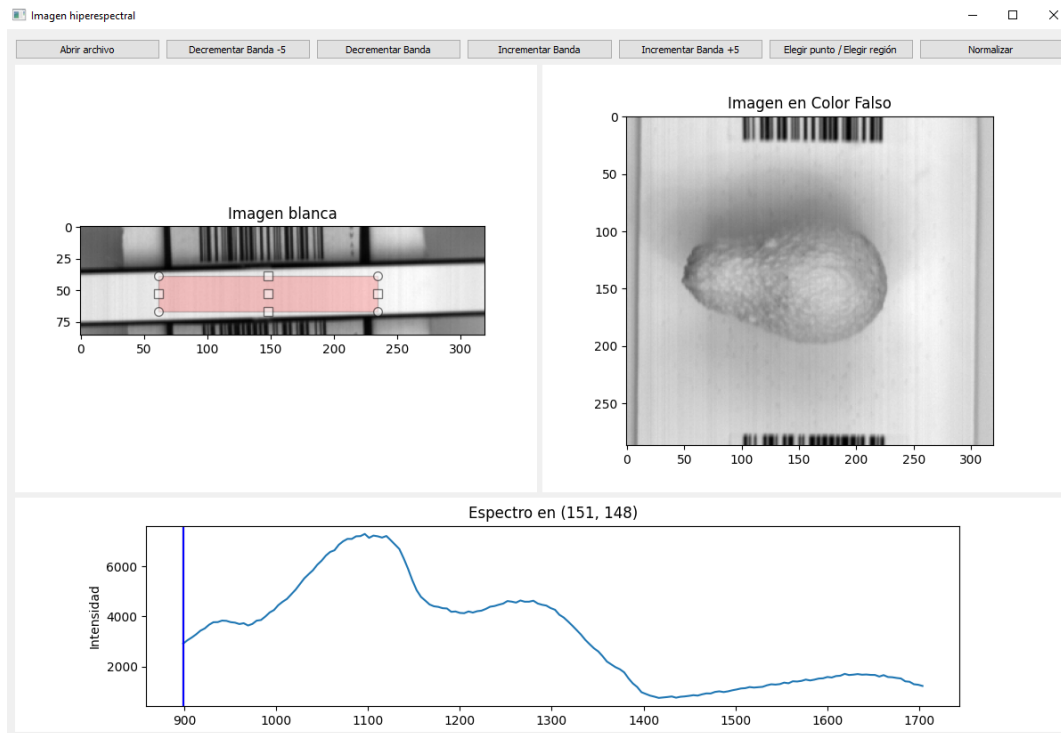


Figura 38 Captura de la selección del área de la imagen de referencia blanca

Una vez realizada la selección por parte del usuario, la aplicación realiza las operaciones necesarias para realizar la calibración. El resultado se puede visualizar a continuación en el gráfico inferior, donde las intensidades se han reducido a valores entre 0 y 1 y corresponde con el espectro de reflectancia que refleja el objeto bajo estudio.

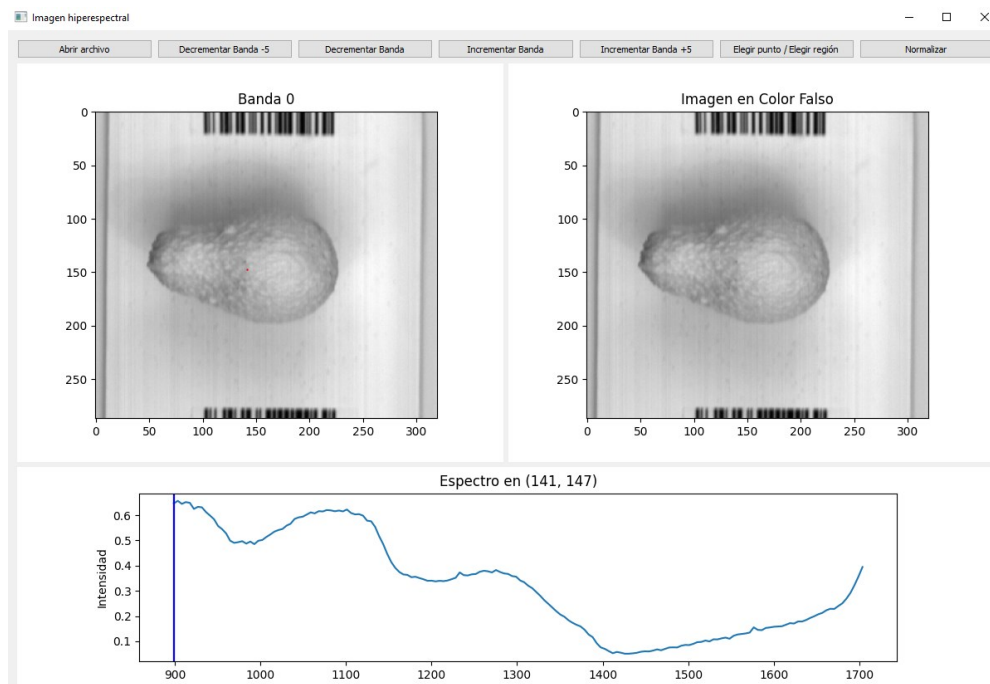


Figura 39 Captura de la aplicación tras la realización de la normalización

10 Conclusiones

Los objetivos principales que se buscaba alcanzar con esta aplicación eran tres:

1. Se pretendía ofrecer una aplicación de visualización y trabajo con imágenes hiperespectrales.
2. Se pretendía que dicha aplicación resultara competente y fácil de usar.
3. Se pretendía que dicha aplicación fuera de código abierto y resultase sencilla de modificar.

Tras la finalización de este primer periodo prolongado de desarrollo, parece claro que se han logrado los tres objetivos.

Para un usuario inexperto en el uso de estas aplicaciones, resulta difícil emitir un juicio sobre las funcionalidades ofrecidas, pero, tras haber revisado el funcionamiento de otras aplicaciones similares, y tras mostrarla a personas con mayor experiencia en este sentido, parece claro que los resultados han sido satisfactorios para ese perfil de usuario, y que los cambios y mejoras a implementar en el futuro se harán sobre una base muy sólida.

En relación al campo de la programación, el éxito también es claro. La aplicación tiene un código bien estructurado, y las variables están inicializadas de forma correcta; ambos hechos favorecen su modificación, y en la experiencia obtenida durante este desarrollo, rara vez se necesita modificar una función previa para poder acomodar nuevas funciones.

Como apreciación final, pese a que por el momento el código no es público, sí se ha comprobado que resulta sencillo de modificar, por lo que, a la espera de ver cómo trabajan con ella usuarios noveles, se puede considerar que este objetivo también ha sido exitoso.

Aun sin ser esta la versión final, es una versión satisfactoria, que deja muchas oportunidades para ser mejorada en el futuro.

11 Trabajo futuro

El desarrollo de la aplicación no ha finalizado tras el sprint 8. Si bien la actual es una versión de la aplicación funcional, no es en absoluto final. Existen multitud de mejoras posibles, junto a requisitos que finalmente no han podido cumplirse aún, y esta versión no es en absoluto final. A continuación, se enumeran algunos de los cambios que se espera poder realizar en el futuro:

- Se deberán mejorar las funciones que buscar permitir la segmentación de la imagen. Esto incluye tomar la decisión de tomar el código previo y hacer que funcione en la actualidad, o escribir funciones totalmente nuevas para tal fin.
- Se deberá ofrecer la posibilidad de guardar algunos de los datos más importantes de la imagen, como la media, mediana, desviación estándar o diversos percentiles, en un fichero externo que podrá consultarse sin abrir la aplicación.
- Se deberá permitir al usuario seleccionar un área de la imagen dibujada libremente, en lugar de limitada a un punto o un rectángulo.
- En relación a la interfaz, se deberá hacer que su uso sea más cómodo, y preferiblemente, emplear iconos en lugar de sólo texto para los botones.

12 Relación del proyecto desarrollado con los estudios realizados

Existen multitud de diferencias entre un ingeniero informático y un programador convencional, pero quizá la más importante sea el hecho de que el ingeniero debe tener la capacidad de resolver cualquier tipo de problema, en lugar de limitarse a la propia codificación. Además, un ingeniero debe ser capaz de efectuar todos los pasos necesarios del desarrollo de un proyecto, siendo, a efectos prácticos, capaz de desarrollar individualmente todo un proyecto sin problema.

El desarrollo de esta aplicación ha requerido del uso de Python, lenguaje de programación que, si bien se ha visto a lo largo del grado, no ha sido un lenguaje en que se profundizase en exceso, y de conocimientos sobre imágenes hiperespectrales, un campo totalmente ajeno al temario aprendido durante el periodo de estudios en la UPV, además de ser realizado individualmente. Sin embargo, precisamente por la preparación obtenida durante el grado, superar todos estos inconvenientes fue viable, y el desarrollo fue en casi todo momento de acuerdo a lo esperado. El conocimiento de las bases de multitud de lenguajes de programación permite aprender a utilizar uno con el que se esté poco familiarizado con un periodo muy breve de preparación; la documentación sobre imágenes hiperespectrales no ha sido más que un paso estándar que se debería llevar a cabo al iniciar cualquier proyecto; y la inversión de una buena cantidad de tiempo permitió sobrellevar el problema del desarrollo en solitario del proyecto.

La conclusión es que, si bien este proyecto se aleja bastante de las situaciones habitualmente vistas a lo largo del grado, en que la mayoría de proyectos se realizaban con cierta supervisión y ayuda técnica, se relacionaban con temas vistos en clase, y se realizaban en equipo, el conocimiento obtenido fue clave para la correcta realización del mismo.

13 Bibliografía

- [1]. González, E. G. (2006). Guía básica de conceptos de radiometría y fotometría. *Universidad de Sevilla, Sevilla*.
- [2]. Manolakis, D., Marden, D., & Shaw, G. A. (2003). Hyperspectral image processing for automatic target detection applications. *Lincoln laboratory journal*, 14(1), 79-116.
- [3]. “El espectro electromagnético: la longitud de onda”. Enlace: <https://cifpn1.com/electronica/?p=4034>
- [4]. “Entendiendo el color (1) – Visión humana”. Enlace: <https://quecamaradefotos.com/camaras/caracteristicas-tecnicas/entendiendo-el-color-vision-humana/>
- [5]. “Cámaras Hiperespectrales”. Enlace: <https://bcnvision.es/blog-vision-artificial/camaras-hiperespectrales/>
- [6]. “Tecnología de imagen hiperespectral en calidad y seguridad alimentaria”. Enlace: <https://itcl.es/blog/tecnologia-de-imagen-hiperespectral-en-calidad-y-seguridad-alimentaria/>
- [7]. “Hyperspectral Viewer”. Enlace: <https://es.mathworks.com/help/images/ref/hyperspectralviewer-app.html>
- [8]. “SpecimINSIGHT”. Enlace: <https://www.specim.com/products/speciminsight/>
- [9]. “HYPER-Tools version 3.0”. Enlace: [HYPER-Tools \(hypertools.org\)](https://www.hypertools.org/)
- [10]. Conesa, A. Imagen del Instituto Valenciano de Investigaciones Agrarias – IVIA. Enlace: https://lh3.googleusercontent.com/p/AF1QipPteJum1Qws8M_kHaru7l-qVjoAFDbrLZbah3za=s680-w680-h510
- [11]. “Visual Studio 2022”. Enlace: <https://visualstudio.microsoft.com/es/>
- [12]. “Microsoft Visual Studio”. Enlace: https://es.wikipedia.org/wiki/Microsoft_Visual_Studio
- [13]. “¿Qué es Python?”. Enlace: <https://aws.amazon.com/es/what-is/python/>
- [14]. “Python”. Enlace: <https://es.wikipedia.org/wiki/Python>

- [15]. “os — Miscellaneous operating system interfaces”. Enlace: <https://docs.python.org/3/library/os.html>
- [16]. “Rasterio: access to geospatial raster data”. Enlace: <https://rasterio.readthedocs.io/en/stable/>
- [17]. “La librería Matplotlib”. Enlace: <https://aprendeconalf.es/docencia/python/manual/matplotlib/>
- [18]. “Rectangle Selector”. Enlace: https://matplotlib.org/3.1.1/gallery/widgets/rectangle_selector.html
- [19]. “Plotting with Matplotlib”. Enlace: <https://www.pythonguis.com/tutorials/plotting-matplotlib/> [17]
- [20]. “NumPy”. Enlace: <https://numpy.org> [18]
- [21]. “PyQt5 - Comprehensive Python Bindings for Qt v5”. Enlace: <https://pypi.org/project/PyQt5/>
- [22]. “PySide6.QtWidgets”. Enlace: <https://doc.qt.io/qtforpython-6/PySide6/QtWidgets/index.html>
- [23]. “QFileDialog”. Enlace: <https://doc.qt.io/qtinstallerframework/scripting-qfiledialog.html>
- [24]. “QMessageBox Class”. Enlace: <https://doc.qt.io/qt-6/qmessagebox.html>
- [25]. “Image processing in Python”. Enlace: <https://scikit-image.org> [22]
- [26]. “skimage.filters”. Enlace: <https://scikit-image.org/docs/stable/api/skimage.filters.html> [23]
- [27]. “**skimage.morphology**”. Enlace: <https://scikit-image.org/docs/stable/api/skimage.morphology.html> [24]
- [28]. “Multidimensional image processing ([scipy.ndimage](https://docs.scipy.org/doc/scipy/reference/ndimage.html#module-scipy.ndimage))”. Enlace: <https://docs.scipy.org/doc/scipy/reference/ndimage.html#module-scipy.ndimage>
- [29]. “CV2: Guía maestra OpenCV para desarrolladores de Python”. Enlace: <https://konfuzio.com/es/cv2/> [26]
- [30]. “time — Time access and conversions”. Enlace: <https://docs.python.org/3/library/time.html>

- [31]. Nobuyuki Otsu: *A threshold selection method from grey level histograms*. In: *IEEE Transactions on Systems, Man, and Cybernetics*. New York 9.1979, S.62–66. [ISSN 1083-4419](#)

14 ANEXO 1

OBJETIVOS DE DESARROLLO SOSTENIBLE

Grado de relación del trabajo con los Objetivos de Desarrollo Sostenible (ODS).

Objetivos de Desarrollo Sostenibles	Alto	Medio	Bajo	No Procede
ODS 1. Fin de la pobreza.				X
ODS 2. Hambre cero.	X			
ODS 3. Salud y bienestar.	X			
ODS 4. Educación de calidad.				X
ODS 5. Igualdad de género.				X
ODS 6. Agua limpia y saneamiento.				X
ODS 7. Energía asequible y no contaminante.				X
ODS 8. Trabajo decente y crecimiento económico.				X
ODS 9. Industria, innovación e infraestructuras.	X			
ODS 10. Reducción de las desigualdades.				X
ODS 11. Ciudades y comunidades sostenibles.				X
ODS 12. Producción y consumo responsables.				X
ODS 13. Acción por el clima.				X
ODS 14. Vida submarina.				X
ODS 15. Vida de ecosistemas terrestres.				X
ODS 16. Paz, justicia e instituciones sólidas.				X
ODS 17. Alianzas para lograr objetivos.	X			

Reflexión sobre la relación del TFG/TFM con los ODS y con el/los ODS más relacionados.

La agenda 2030 debería ser el principal punto de referencia para la mayor parte de proyectos informáticos y empresariales en el futuro. Firmada en 2015 por la Organización de las Naciones Unidas, contiene 17 Objetivos de Desarrollo Sostenible, u ODS, que deberían ser alcanzados antes del final de esta década, y que, como principal característica, deben ser perseguidos por absolutamente todos los colectivos de la sociedad, incluyendo obviamente a gobernantes, pero también a todos los ciudadanos y trabajadores de cada país. Teniendo esta información presente, y partiendo de la base de que también el trabajo de los ingenieros informáticos debería acercarnos al cumplimiento de esos objetivos, a continuación, se exponen algunos de los ODS relacionados que este proyecto podría ayudar a alcanzar y que puede servir para promover.

- Objetivo 2: Hambre cero. El segundo objetivo se relaciona con la búsqueda de la reducción y finalmente eliminación de las desigualdades en materia alimenticia y de nutrición, poniendo fin al hambre, y tomando también medidas contra el sobrepeso. El trabajo con imágenes hiperespectrales realizado en este proyecto tiene como uno de sus beneficios principales el de poder mejorar el tratamiento de alimentos de origen natural, permitiendo identificar malformaciones y problemas que pudieran ser perjudiciales para la salud. De ser empleada la aplicación de forma correcta, podría ser empleada para mejorar la calidad de una considerable cantidad de alimentos en el futuro, así como para impedir el desperdicio de alimentos en buen estado, un problema considerable en la actualidad.
- Objetivo 3: salud y bienestar. Este objetivo plantea la necesidad de mejorar las condiciones sanitarias de la población. Partiendo de lo que ha sido planteado al desarrollar la relación del proyecto con el objetivo anterior, se puede afirmar que la aplicación resultante de este trabajo podría ayudar directamente a mejorar la calidad de vida de muchas personas, al mejorar la calidad de su alimentación.
- Objetivo 9: Industria, innovación e infraestructuras. El noveno objetivo es quizá el que más interesante resulta para los ingenieros, y para este trabajo en particular. La labor de los ingenieros como colectivo para ayudar al desarrollo correcto de este objetivo debería pasar por trabajar en tecnologías que funcionen y sean desarrolladas de forma sostenible, generando tecnologías duraderas e innovadoras. Dado que uno de los objetivos de este trabajo pasa por hacer que sea sencillo para cualquier usuario darle uso y ampliarlo o mejorarlo en el futuro, y que el acceso al

mismo será gratuito y abierto, este objetivo encaja perfectamente con lo que se ha buscado desarrollar.

- Objetivo 17: Alianzas para lograr objetivos. Como se ha mencionado varias veces previamente, uno de los puntos más destacables de esta aplicación será el hecho de que es de código abierto, por lo que la cooperación entre usuarios y programadores podría ser clave en su futuro. Teniendo esto en cuenta, y habiendo comprobado que también varios de los objetivos anteriores a este encajan con este proyecto, podría considerarse esta aplicación como una de esas alianzas buscadas, con el objetivo de promover los ODS y de crear un futuro mejor y más sostenible para todos.